Applied Project
Audio

Zohar LasKar Koriat
Achituv Drot
Yoel Graumann

Abstract:

In Data Science, a prediction task is an Endeavor involving building a model that can make predictions and inferences based on a given dataset. The goal is to learn patterns from the training data which can then be used to make predictions for new, unseen data. We received the audio Dataset and attempted to learn the patterns with two different supervised machine learning models: Lasso and XGBoost. We found out that Model Lasso works better than XGBoost on both datasets. We tried to balance our data by the MixUp data augmentation technique, and we were disappointed to find that the MixUp does not improve the predictions of the model.

Introduction

We received the Audio Modality for our prediction project. The initial Dataset, created by Itamar, contained 1920 features and 6423 samples, which were extracted from the output of the embedding models. Itamar's model for creating the dataset was simply extracting the following percentiles (0.1,0.25,0.5,0.75,0.9) from the output of the embedding models, which originally had almost 4 million samples. Most of the features seem to follow a normal distribution, with a few following a bimodal distribution.

The response variables reflect the level of emotion exhibited by participants in each segment. There are a total of 43 different emotions, and after looking at their distributions we found that there are 5 types of different distributions in our database. Therefore we chose to analyze one emotion from each type of distribution. Note: the emotions arousal, valence and interest has different distributions than the rest of the emotions and therefore they are 3 of the 5 distributions.

Following that we decided to focus on arousal, valence, interest, despair and joy. Most of the distributions of the emotions are right skewed, with valence and arousal being the only symmetric distributions. The emotions have been normalized by Itamar, and we used them in our prediction task.

In this project, the goal is to attempt to improve the prediction accuracy across the five distinct response variables listed above, by leveraging Itamar's dataset[1], our dataset[2], and our augmented datasets[3]. The Main goal is to reduce the normalized (by SD) Root mean squared error. We aim to improve the prediction accuracy by experimenting with two approaches. The first is by using a linear model, specifically Lasso regression, which works by selecting the most important features and reducing overfitting. The Second is by using a

[1] This is the database with Itamar's aggregation.
[2] This is the database with our aggregation.
[3] This is the database with our aggregation with MixUp data augmentation technique.

non-linear model, XGBoost, which can capture complex interactions between features. By experimenting with these two models, we hope to achieve more accurate and robust predictions.

For our research task, we have decided to experiment with the application of MixUp data augmentation techniques to improve model performance. MixUp works by generating new & synthetic training data by combining two original data points. This method creates new data points which ideally should encourage the model to generalize better. We aim to apply MixUp to two different datasets: Itamar's dataset, and to our own aggregation variant. By introducing this augmentation method, we want to evaluate whether the additional diversity in the training data will lead to improvements in the models' predictive performance. Again, we want to lower the normalized RMSE. Our interest in determining if this augmentation can make the model more robust and able to make more accurate predictions, especially when the original data is imbalanced like our data.

Preliminary analysis of the data

	Mean	Median	Std. Dev	Min	Max	Skewness	Kurtosis
arousal	4.74	4.59	0.89	1.51	7.52	-0.05	0.42
valence	3.64	3.61	1.19	0.70	7.37	0.33	-0.07
interest	3.01	3.39	2.04	-1.38	8.18	-0.20	-0.82
despair	2.07	1.56	2.03	-0.00	7.56	0.73	-0.58
joy	1.02	0.77	1.42	0.00	7.15	1.87	3.32

The following table is the simplest summary statistic of the emotions, considering Itamar's data. For Arousal, the mean and median are closely aligned, suggesting a symmetric distribution. The low standard deviation indicates that most segments cluster around the mean, with minimal extreme values. The near-zero skewness supports the notion of symmetry, and a positive kurtosis suggests a slightly peaked distribution, where values are concentrated near the mean. For Valence, the mean and median are almost the same. However, the higher standard deviation compared to arousal shows more variability in this emotion. The slight positive skewness points to a few higher valence scores, and the kurtosis is close to zero, suggesting a normal distribution with a balanced spread of values. Interest displays more variability, with a high standard deviation and a wider range of emotional intensity, including a negative minimum. The negative kurtosis suggests a flatter, more spread-out distribution compared to a normal curve. Despair shows a slightly skewed distribution, with a mean higher than the median, indicating some positive skewness. The high standard deviation suggests significant variability in the emotion, with a wide range from zero to a maximum of 7.56. The negative kurtosis indicates a flatter distribution, suggesting more evenly spread despair scores without many extreme outliers. Finally, joy has a more strongly skewed and peaked distribution. The mean is slightly higher than the median, but the significant positive skewness indicates that most segments' emotions had lower levels of joy, with a few outliers of high levels. The high kurtosis reflects a sharply peaked distribution, suggesting that most participants' joy scores are concentrated around the minimum, with only a few extreme outliers. In summary, arousal and valence are more symmetric, while interest, despair, and joy demonstrate more variability and skewed distributions.

There are 1920 features, with most of the feature means around 0.44. The standard deviations range from 0.05 to 0.47, suggesting a mixture of low to moderate variability across features. The minimum values range from -4.17 to -2.99, while maximum can be as high as 4.16. The IQR for most features lies between 0.16 and 0.62, indicating that the central 50% of the features is well concentrated, while the presence of extreme minimum

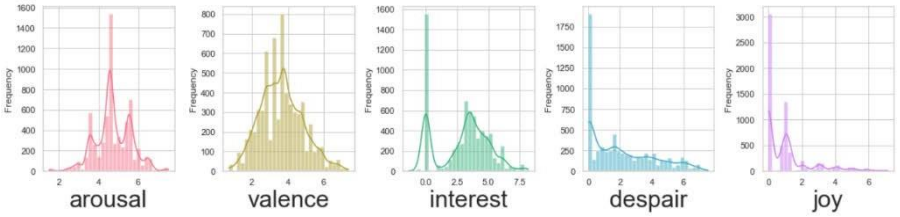
and maximum values suggests some features may have outliers or long tails in their distributions. In the following table we performed a simple correlation analysis and we reveals that valence has a strong influence on emotions, showing a significant positive correlation with joy and a strong negative correlation with despair. This indicates that as valence increases, joy rises while despair decreases, underscoring the difference between positive and negative emotions. Interest and arousal are weakly correlated with the other emotions. The relationship between interest and despair is almost non-existent, indicating that these emotions do not influence each other. Overall, valence appears to be the key emotion shaping the overall emotional dynamics. Using hierarchical clustering, valence and joy are identified as the most closely related emotions. In contrast, arousal and interest group together, indicating they represent a distinct emotional dimension separate from the valence-joy-despair cluster.[i] We also performed Association Rule Mining on the emotions and found several patterns, we kept only the rules which had a confidence of above 0.75, and we defined “low” emotions to be less than the median, and “High” emotions to be above or equal to the median. A strong relationship was found between high arousal and low valence, which often led to high levels of despair, which might indicate that negative emotional arousal is often accompanied by feelings of despair. Additionally, low arousal and low despair were strongly associated with high valence, suggesting that being calm is linked to positivity. High interest combined with low despair was found to predict high valence with 78% confidence, underscoring that interest is generally associated with positive emotions. Another rule that we found was that high joy and low despair led to high valence, 84% of the time. Interestingly, the combination of high interest and low valence was associated with high despair in 77% of the cases, suggesting that interest can coexist with negative emotions. Overall, valence seems to be a central emotion in our data, influencing both despair and joy across multiple patterns, with despair and valence acting as opposites.

	arousal	valence	interest	despair	joy
arousal	1.000000	0.023847	0.106272	0.111831	0.138984
valence	0.023847	1.000000	0.130614	-0.512874	0.530522
interest	0.106272	0.130614	1.000000	0.018450	0.247375
despair	0.111831	-0.512874	0.018450	1.000000	-0.261905
joy	0.138984	0.530522	0.247375	-0.261905	1.000000

Methods

In this prediction task, we were provided with a dataset containing 1920 features and 43 different target variables, namely emotions. The features were extracted from the output of embedding models, capturing various representations and relationships within the data. The goal was to build a model capable of predicting the emotion values while minimizing the root mean square error (normalized by SD), which essentially measures the average magnitude of the error between predicted and actual values. One of the key goals is not only to develop an accurate predictive model but also to outperform the LSTM model that used by Itamar, setting a benchmark for improved performance in terms of “normalized RMSE”.

We focused on the following emotions: Arousal, Valence, Interest, Despair, and Joy due to their unique distributional characteristics. The following graphs show the distribution of these emotions.



Arousal and Valence are fairly balanced, with slight deviations from a normal distribution, making them suitable for standard predictive modelling. Interest and Despair, display mild skews and flatter distributions, suggesting more spread

out data, which can pose interesting challenges for prediction. Finally, Joy is highly skewed, with most values concentrated at the lower end and a few extreme outliers, offering a more complex distribution to account for. These variations provided a meaningful reason to investigate these emotions, allowing for a deeper exploration of how our model handles different types of distributions. In the preprocessing step of our project, we first scale the features by standardizing them, which involves removing the mean and scaling each feature to unit variance. Secondly, we apply PCA to reduce the dimensionality of the data, keeping only the components that explained 99.99% of the variance. This allowed us to significantly reduce the feature space while preserving the essential information. The transformed features were then fed into the model for training. Note that we split the data into an 80% train 20% test split. For model validation, we used stratified K-fold validation, like what Itamar did. However, while Itamar treated the data as a temporal dataset for his stratified K-fold, we approached the validation by treating each row as a separate sample. So, although both methods aimed at ensuring a balanced distribution of the target variables in each fold, our approach did not incorporate temporal dependencies, making the validation methods very similar, but not the same. The success indicator we decided to use was the root mean square error (RMSE), normalized by the standard deviation. The different emotions, as we have showed above, have varying levels of variability. Normalizing RMSE by the standard deviation is a critical step because it enables a fair comparison of the models' performance across the different emotions. Without normalization, RMSE values could be difficult to interpret, especially when the emotions have different spreads or ranges and simply comparing their "raw RMSE" will be misleading. For example, if one variable has a narrow distribution, a lower RMSE might still represent poor performance relative to that variable's range. This Normalized RMSE is a success indicator because it provides a standardized measure of how well the model performs relative to the variability in the data. By normalizing the RMSE, we ensure that the performance metric is scale independent. A lower normalized RMSE indicates that the model's predictions are closer to the true values relative to the variability of the emotion, making it a meaningful indicator of success. We will be using Itamar's Stacked LSTM model as our base model. This model consists of two layers, the first has 256 units, and the second has 128 units and was trained for 50 epochs. Our aim is to use Itamar's model as a starting point and attempt to improve upon its performance with our own models. We tested two Different models, Lasso Regression and XGBoost. Lasso regression is a linear model that uses L1 regularization, which helps prevent overfitting by penalizing large coefficients. It shrinks the less important feature coefficients to zero, keeping the important features with nonzero coefficients. XGBoost on the other hand, is a nonlinear, decision tree based ensemble method. It works by building multiple decision trees in a sequential manner. Each new tree tries to correct the errors made by its predecessor. In short, this model is effective at capturing complex and non-linear patterns in the data. It might have the upper hand in cases where a simple linear model, like lasso regression, might struggle. As for the new aggregation, we wanted to expand on Itamar's aggregation, and instead of using 5 percentiles, we decided to use the following percentiles (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9). Additionally, we took the mean, standard deviation, max and min. focusing on 13 statistics in total, ensures that the model has a richer set of features to work with, ultimately leading to more accurate predictions by better reflecting the data's underlying patterns. While we gain greater detail and resolution, we know that we might have redundancies, especially if the additional features are not informative or relevant to the target variables.

The explanation for describing the method we used for our research task: since in most of the observations in our data the emotions are not present, we would like to create samples that the emotions will be more present and as a result the model will be able to learn better than before the range of possibilities of the emotions and finally it will be able to predict better for new data. And that's why we chose to use the MixUp black-box data-augmentation technique to potentially improve the accuracy, the robustness and the generalization of the models. Mixup generates new samples by combining pairs of existing training samples.

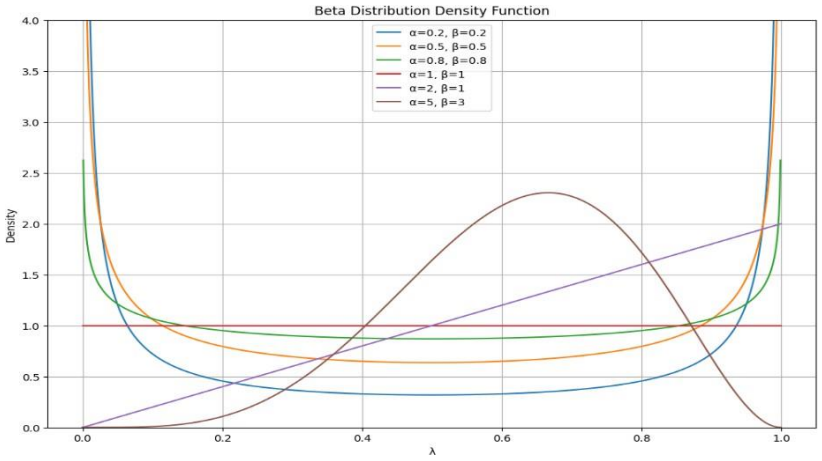
The algorithm of the MixUp method:

1. **Random Pairing:** Randomly select pairs of examples (x_i , y_i) and (x_{jj} , y_{jj}) from the training data.
2. **Interpolation:** Create a new example by linearly interpolating between these pairs. The new example (x_{nenn} , y_{nenn}) is defined as:

$$x_{nenn} = \boldsymbol{\lambda\lambda} x_i + (1 - \boldsymbol{\lambda\lambda}) x_{jj}$$
$$y_{nenn} = \boldsymbol{\lambda\lambda} y_i + (1 - \boldsymbol{\lambda\lambda}) y_{jj}$$

where λ is a mixing coefficient sampled from a Beta distribution $Beta(\alpha , \beta)$.

Here is the visualize how changing the parameters of the Beta distribution affects its density:



From everything that was said above and looking at the above graph, for our database we choose λ~Beta(2,1), this will produce samples that are close to one of the original data points.

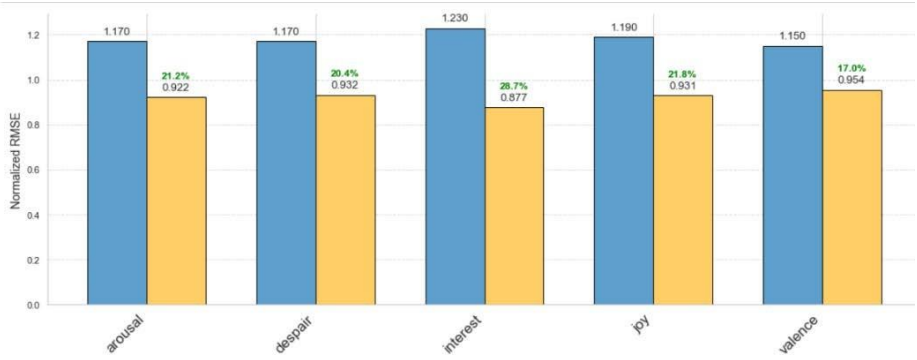
Because one of the significant challenges in our dataset was that most of our samples have no emotion observed (the value 0 in the emotions columns) in our training data, we would like to correct this imbalance by generate more samples for the training data in which an emotion was observed in order to produce more balance training data set. As a result we would like to use data augmentation on samples that the emotions are more present. to address this issue, we developed a specialized version of MixUp, where the pairs of examples (x_i , y_i) and (x_{jj} , y_{jj}) are not randomly selected from a uniform distribution but instead they were weighed down by exponential weights. The weighting function we applied is: $\frac{e^{y_i * weight}}{\sum_{jj=0}^{N} e^{y_{jj} * weight}}$,

We use these weights to randomly select two distinct samples, giving preference to those with higher emotion values. This selection process is part of a data augmentation method that aims to mix and generate new synthetic data points that have high emotion values.

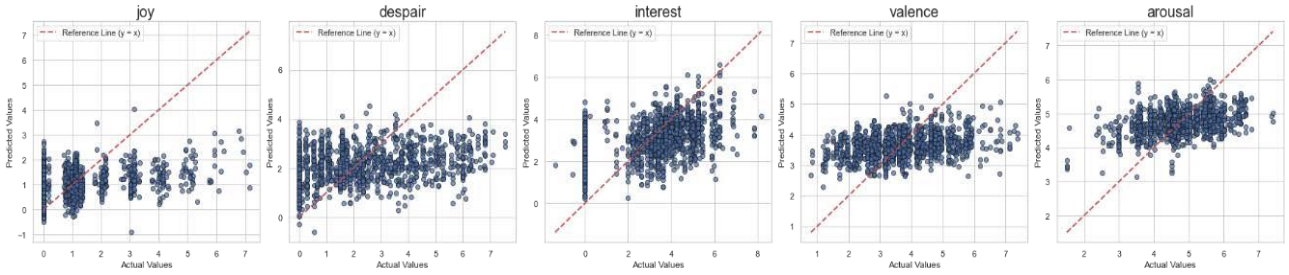
As the value of the weight increases, the probability of selecting samples becomes more concentrated around those with higher emotion values. For all the reasons stated above, we create 3,000 new training data points and we try the following weight values: $weight = 0,1,2,3,4,6,10$. At weight = 0, all samples are equally likely to be selected, while at higher weights (e.g., 6 or 10), only samples with significantly high emotion values dominate the selection process. Therefore we will experiment with different values of weight.

Note: we performed this MixUp technique for each emotion class individually.

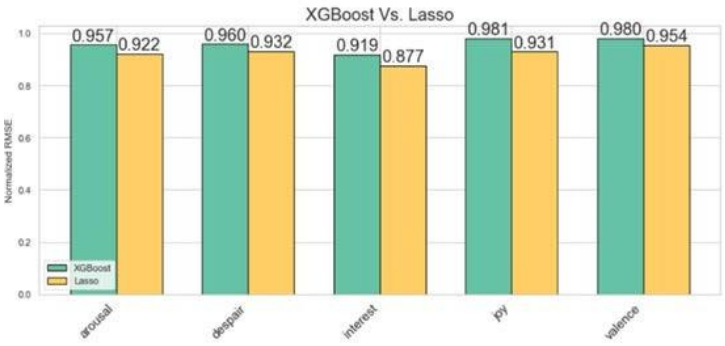
Results[ii]



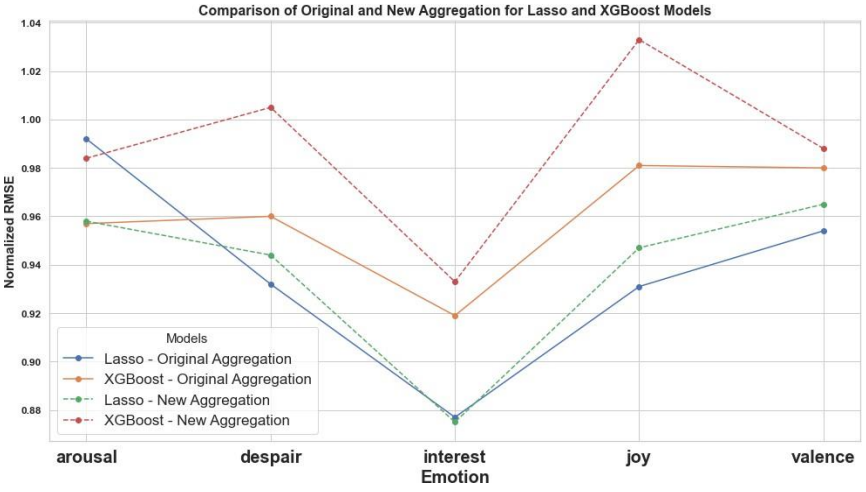
The above is a comparison of Itamar's LSTM Model (Blue) and Our Lasso Regression Model (Gold). Above each bar chart there's the Normalized RMSE. In addition, above each golden barchart, there's the percentage improvement of Lasso over the LSTM Model in green. This Comparison reveals that the Lasso Regression model consistently outperforms the LSTM model. For Arousal, the Lasso model achieves a 21.2% improvement, reducing the RMSE from 1.170 to 0.922. Similarly, in Despair, the Lasso model shows a 20.4% improvement, lowering the RMSE from 1.170 to 0.932. The most notable gain is seen in predicting Interest, where the Lasso model improves the RMSE by 28.7%, from 1.230 to 0.877. Joy and Valence also see substantial improvements of 21.8% and 17.0%, respectively. This highlights that the Lasso Regression model delivers better performance than the LSTM model across all emotions, with its largest improvement in Interest and the smallest in Valence. In the following graphs we performed a very simple error analysis, where we plotted the predicted vs real values, to see if our models capture the relationships and patterns in the emotions.



In nearly all cases, the Lasso model shows a bias toward mid-range values, predicting within a narrow range regardless of the actual values. This indicates that the model regresses toward the mean and fails to capture the full variability, especially at the extremes. Additionally, the model struggles to predict both low and high values accurately. This consistent underprediction of high values and, in some cases, overprediction of low values suggests underfitting, likely due to the regularization applied by Lasso. This aggressive regularization can oversimplify the model, leading to underfitting, especially when the relationships in the data are complex or non-linear. This underfitting manifests as the model failing to predict extreme values, both low and high. Because of these weaknesses, we decided to use XGBoost. This model, with its tree-based structure and iterative boosting process, is well-suited to handle non-linearities, capture variability across a wide range of values, and improve prediction accuracy for extreme values by learning from previous errors. This makes XGBoost a more flexible and robust alternative, capable of addressing the issues that limited Lasso's performance, Or so we thought. Below is a graph showing the comparison between Lasso and XGBoost.



XGBoost did outperform Itamar's LSTM model, but it did not outperform the Lasso regression model. XGboost did, in fact, almost perform as good as lasso but not quite, as can be seen in the "XGBoost Vs. Lasso" figure. We Performed a simple error analysis and found that XGBoost had very similar plots to what we showed for Lasso. In other words, XGBoost also had trouble with under/over predictions, depending on the circumstances, as explained above. Note that one of the reasons XGBoost did not perform as expected, we believe, was because we decided to train our models locally and did have finite resources. We believe that if we were to train the models on a more exhaustive list of hyperparameters, we might've had better results than the Lasso. One thing we noticed was that the two models' performance was at their peak, relatively speaking, when the predicted emotion was interest. We conducted an importance analysis for both the XGBoost and Lasso models, specifically for the 'interest' emotion. Among the top 50 most important features from each model, we identified 4 features that were considered important by both models. One of these features was of the 0.75 percentile, while the remaining three were of the 0.9 percentile (from Itamar's aggregation). This suggests that extreme or larger values within the distribution are critical for capturing patterns for 'interest', where both models do best. We ran the two models on our custom-made aggregated data, as explained in the "Methods" section and compared all the results so far in the following graph. The Lasso model showed minor changes between the original (Itamar's aggregation) and new (our aggregation) aggregations, with slight improvements in arousal and stable performance for Interest, but small declines in Despair, Joy, and Valence. In contrast, the XGBoost model experienced



consistent performance declines across all emotions on the data with the new aggregation, with noticeable deteriorations particularly in Despair and Joy. This suggests that XGBoost is more sensitive to data representation and may require additional hyperparameter tuning. Overall, Lasso demonstrated more stability, while XGBoost was more sensitive to the new aggregation approach.

The results of the research task: We tried the following weight values: 0,1,2,3,4,6,10 on the MixUp data-augmentation technique with Lasso model for Itamar aggregation and for our aggregation. For our aggregation, the best results of the Normalized RMSE were obtained when the value of the weight was 10 and for Itamar's aggregation, the best results obtained when the value of the weight was 0. In both aggregations the worst results of the Normalized RMSE were obtained when the values of the weight was 1-4.

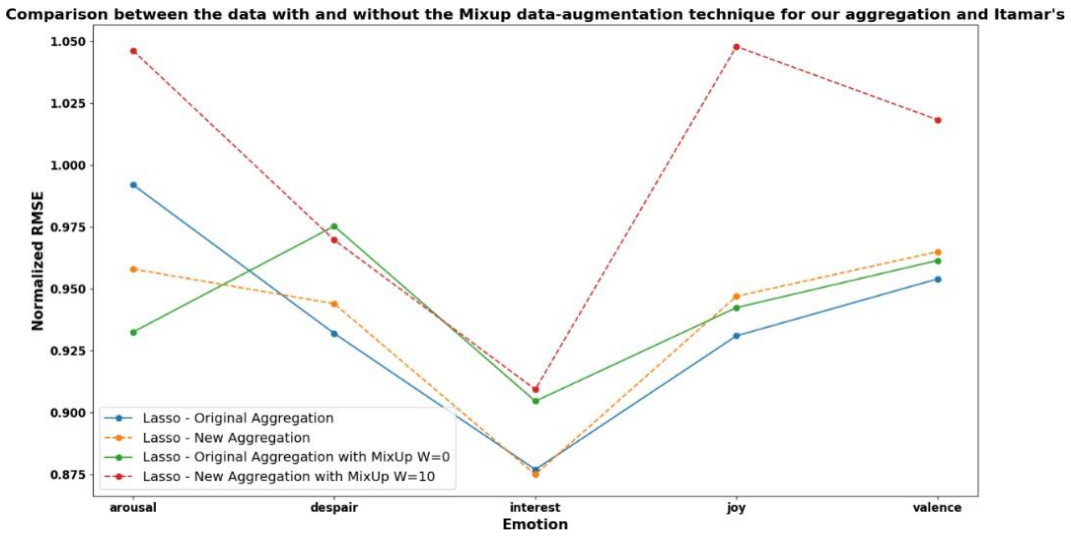
When weight = 0 it represents standard MixUp without weighting, so all samples have an equal chance of being selected, regardless of their emotion values and when weight=10 the selection becomes highly skewed, where only samples with the highest emotion values are likely to be chosen so samples with lower emotion values will have a negligible chance of being selected.

In our aggregation, we included more percentiles (0.1 to 0.9 in 0.1 increments), and additional statistics such as the standard deviation, which captures more granular information about the data's distribution and variability. With more detailed and complex features, emphasizing stronger signals via higher weights (weight = 10) helps the model to better capture useful patterns, especially when the key information in our aggregation is more concentrated in extreme or distinct emotion values. High weight values focus on these extremes, which are beneficial in our aggregation where important distinctions are likely to be captured in outliers or specific patterns reflected by these advanced statistics. As a result it captures the complexity of the data more effectively, resulting in better performance.

For Itamar's aggregation, which involves fewer percentiles (0.1,0.25,0.5,0.75,0.9), this uniform weighting (weight = 0) likely helps create more generalized and stable new examples for training. The reduced complexity of the features in this aggregation might make the uniform sampling work better, as it prevents overemphasis on extreme or specific examples.

Intermediate weighting (weight = 1, 2, 3, 4) leads to a suboptimal balance, where neither the general diversity of the dataset is maintained (as in uniform weighting) nor the most important data points are prioritized (as in high weighting) leading to worse results in both cases.

Below is a graph that shows our final results for the Lasso model that gave us the best results with Itamar's aggregation and with ours, with and without the MixUp data-augmentation technique with the weights we wrote above that gave the best results for each aggregation:



We can see that for the vast majority of cases, the addition of data using the MixUp method does not improve the Normalized RMSE. There are several possible reasons for this result that are related to the way the MixUp method works and the nature of the data:

- The original aggregation and our aggregation, might already capture enough information from the data, like the underlying patterns in the data, especially because these aggregations contain percentiles and our aggregation also contain min, max, standard deviation and mean statistics, which can make MixUp less effective. For instance, using percentiles like 0.1, 0.5, and 0.9 gives a good spread of the data distribution. In such cases, adding more synthetic data points through MixUp may not provide additional useful information, and might even degrade the model's performance by introducing noise. And this might explain why MixUp with w=0 performs better, while introducing weighted MixUp leads to worse performance.
- Lasso regression performs well when the data is sparse and when there is a clear signal in the feature space. By introducing additional mixed data points, MixUp can smooth the data, which could reduce the sparsity or introduce noise. This might make it harder for Lasso to select the most important features, resulting in higher errors. The success of MixUp is often seen in more complex models like deep neural networks, where regularization and overfitting are major concerns. For simpler models like Lasso, which already incorporate regularization (through the L1 penalty), the additional regularization effect of MixUp may not provide much benefit and can sometimes hurt performance.

Summary and discussion:

In this project, we tried to improve the prediction accuracy of the emotion response variables using two distinct supervised machine learning models: Lasso Regression and XGBoost. We experimented with both Itamar's original dataset and our custom aggregation, incorporating a range of percentiles and statistics to capture data patterns more effectively. While our Lasso model consistently outperformed both Itamar's LSTM model and XGBoost across all target emotions, we encountered various challenges and limitations throughout the process. Our approach of examine two models, one linear and one non-linear highlighted important trade-offs between model simplicity and complexity. Lasso, with its inherent feature selection, was stable and effective, outperforming XGBoost despite the latter's ability to

capture non-linear patterns. However, neither model could predict extreme values accurately, indicating potential underfitting. Our attempt to enhance the models through the MixUp data augmentation technique yielded mixed results. The method showed limited improvement and, in some cases, degraded performance. This suggests that the existing aggregations already captured sufficient information, and the addition of synthetic data points introduced noise rather than helpful variation. We learned a lot from this project. We learned that sometimes the more simple model actually performs better than the more complex model, especially when computational resources are limited. We learned that data augmentation does not always necessarily work. While mixup has been proven to be effective in some deep learning models, it did not significantly benefit our simpler model. Additionally, we recognize that although XGBoost did not outperform lasso in this project, we believe that if we had better resources we could have had better performance with XGBoost. This shows that we need exhaustive hyperparameter tuning to unlock the potential of the more complex models. The main limitation in this project was actually the dataset itself. In other words, the imbalance in the dataset, with many samples containing minimal emotional expression, made it challenging for our models to capture the underlying representation of the data, particularly for skewed highly skewed emotions. We think that the next logical step would be to explore alternatives for data augmentation. Given the limitation of mixup for our models, a good idea would be to employ techniques like noise injection[iii], this could enhance the training data by creating subtle variations in our original features and preserving the underlying structure of the data. Another interesting next step would be to incorporate a more exhaustive list of hyperparameters to tune the XGBoost with. In conclusion, while Lasso regression surprisingly proved to be the most effective model in this project, our experience underscored the need for careful consideration of data augmentation techniques, resource planning and hyperparameter tuning. We hope that the insights presented in this project will help future researchers in this area.

[i] https://github.com/YoelGraumann/Applied-Project-code/blob/main/Applied%20Project%20-%20Association%20Rule%20Mining.ipynb

[ii] https://github.com/YoelGraumann/Applied-Project-code/blob/main/Applied%20Project%20Result%20Analysis.ipynb

[iii] https://www.geeksforgeeks.org/noise-injection-for-training-artificial-neural-networks/