PREDICTING FRUSTRATION FROM HEART-RATE SIGNAL WITH MACHINE LEARNING MODELS

A PREPRINT

Benjamin Banks*
Danish Technical University
s234802@dtu.dk

June, 2024

ABSTRACT

This paper explores how to predict frustrations levels from heart rate signals using machine learning, focusing on generalizability to new individuals. To determine the most effective approach, two layer cross validation, is used to compare Random Forest(RF), Logistic Regression(LR), K-Nearest Neighbors(KNN) and AdaBoost(ADA) and 3 baselines, by using 7 metrics, and two p-value tests. Based on a subset of the EmoPairCompete datasetDas et al. [2024], it can be concluded, that LR had the worst performance. We find indications that KNN have highest performance.

Keywords Machine Learning · Frustration Prediction · Heart Rate Signals · Cross-Validation · Model Generalization

*Code is available at GitHub

1 Introduction

As the world becomes increasingly interconnected, technology continues to change the way we live. One positive aspect of this change is the potential for technology to help us better understand ourselves. Historically, diaries have been a method for logging personal information. With the advancement of technology, recognizing and interpreting emotions can further improve this experience. This paper explores how to predict frustrations levels from heart rate signals using machine learning. To determine the most effective approach various models will be tested.

2 Methods

2.1 Dataset Description

The dataset to be used is a subset of the EmoPairComplete Das et al. [2024] and contains 168 observations, with 11 attributes. The data comprises of repeated frustration measurement. Each individual went through 4 rounds, with 3 phases each, totaling 12 measurements per individual. Each individual was either the puzzler or not and also was in one cohort only. We must consideration the effect of these groupings, when splitting the data for testing and training later. Since only heart rate signals are wanted as features for the predictions, only the following features will be used, HR Mean, HR_Median, HR_std, HR_Min, HR_Max and HR_AUC. The participants self-rated frustration level will be used as target variable. Therefore the attributes, individual, Cohort, Group and Phase are seen as metedata.

2.2 Data Preprocessing

The data preprocessing happened in stages.

The first stage **Binarization of Target Variable** was done by categorizing values of 5 and above as 'frustrated' and values below 5 as 'not frustrated'. This transformation emphasizes whether an individual is frustrated rather than the degree of frustration, and is applied to the entire dataset initially. This led to a class imbalance of 14% frustrated and 86% not-frustrated.

The second stage **Mitigating Class Imbalance and Scaling** was done by first doing Synthetic Minority Over-sampling Technique(SMOTE) as presented in [Bowyer et al., 2011] with k_neighbors=5, only resampling the minority class. k_neighbors=5 was arbitrarily chosen. The reason for over-sampling the training data, is to give the models a better opportunity to understand the minority class.

Subsequently, the mean was removed, and the data was divided with the standard deviation, because the different heart rate signals can vary much in mean and standard deviation, and some models might get impacted by this difference. These preprocessing steps (SMOTE and scaling) were incorporated into a pipeline along with the classifier. This pipeline is rerun each time a classifier is trained to ensure new random data generation, better reflecting performance on new data.

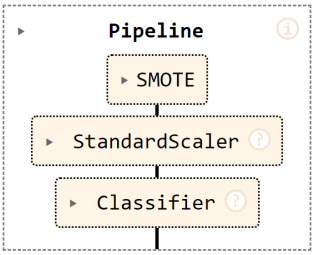

Figure 1: The pipeline applied to each classifier

2.3 Model Selection and Validation

The data is from 14 individual test persons, and we will make sure to use leave one group out by individuals, because we want to generalize to new individuals. 7 different classifiers were tested to complete the pipelines. Four common models, such as Random Forest(RF), Logistic Regression(LR), K-Nearest Neighbors(KNN) and AdaBoost(ADA), which are chosen to increase the chances of efficiently finding a model to solve the problem.

The last three classifiers are baselines, and are stratified dummy classifier (Base[S]), which predicts based on class distribution, a classifier that always predicts the positive class (Base[+]), and one that always predicts the negative class (Base[-]). These baselines provide additional perspectives for comparison.

To optimize the performance of the machine learning models, a grid search over the hyperparameters using two-fold cross-validation was performed, optimizing for the F1 score. The test scores from the outer k folds were used to compare the models, while the inner validation sets were used to conduct the grid search for hyperparameter tuning. The specific hyperparameters and their ranges for each model are detailed below:

- **Random Forest (RF)**:
  - Number of estimators (`n_estimators`): [10, 50, 100, 200, 500, 1000]
  - Maximum depth of the tree (`max_depth`): [2, 5, 10, 20]
- **Logistic Regression (LR)**:
  - Inverse of regularization strength (`C`): [0.001, 0.01, 0.1, 1, 10, 100, 1000]
  - (This range is represented as `np.logspace(-3, 3, 7)`, which generates 7 logarithmically spaced values between 10^{-3} and 10^{3}.)
- **K-Nearest Neighbors (KNN)**:
  - Number of neighbors (`n_neighbors`): [1, 3, 5, 7, 9]
- **AdaBoost (ADA)**:
  - Number of estimators (`n_estimators`): [10, 50, 100, 200, 500]

These parameter ranges were chosen based on common practices to cover a wide range of possible model complexities and regularization strengths. The two-fold cross-validation ensures that the model selection process is robust and generalizes well to new data by using separate inner validation sets for hyperparameter tuning and outer test sets for model evaluation. Because the goal is to generalize to new individuals, the outer cross-validation loop used leave-one-group-out, ensuring that data from the same individual did not appear in both training and validation sets simultaneously. To compare the performances of the models, the following metrics will be used:

F1-Score provides a single metric that balances both the precision and recall, making it useful for evaluating on this imbalanced dataset.

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{1}$$

Balanced Accuracy is the average recall obtained on each class. As F1, it adjusts for class imbalance.

$$BalancedAccuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{2}$$

Precision measures the proportion of true positive predictions among all positive predictions. It indicates how many of the positively predicted cases were actually positive.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall (or sensitivity) measures the proportion of true positive predictions among all actual positive cases. It indicates how many of the actual positive cases were captured by the model.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Negative Predictive Value (NPV) measures the proportion of true negative predictions among all negative predictions. It provides insight into the model's performance on the negative class, and will be useful for comparing models to Base[-].

$$NegativePredictiveValue(NPV) = \frac{TN}{TN + FN} \tag{5}$$

Matthews Correlation Coefficient(MCC) ranges between -1 and 1. 1 represents perfect prediction, 0 same performance as random prediction and -1 total disagreement between predictions and true values. It will be used to compare the models to random guessing. Baldi et al. [2000]

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6}$$

ROC-AUC or The Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) measures the model's ability to distinguish between classes across different thresholds. It is a good complement to all the above, since it does not assume that the class threshold is 0.5.

To statistically compare the models, two test will be used. Cochran's Q test to test for any differences in performance between any of the models. If there is indication of significant difference, a pairwise McNemar test will be performed, which leads to a total 21 comparisons. By Bonferroni correction the usual significant level of 0.05, we get an adjusted significance threshold: $p = \frac{0.05}{21} = 0.0024$

3 Results

3.1 Performance Metrics

Table 1: Classifier Performance Metrics

	Classifier	F1 Score	Balanced Accuracy	Precision	Recall	NPV	MCC	ROC-AUC
0	**RF**	0.1860	0.4792	0.1290	0.3333	0.8491	-0.0302	0.4673
1	**LR**	0.1333	0.3611	0.0833	0.3333	0.7778	-0.1964	0.3142
2	**KNN**	0.2917	0.5903	0.1944	0.5833	0.8958	0.1277	0.5812
3	**ADA**	0.2469	0.5451	0.1754	0.4167	0.8739	0.0667	0.5415
4	**Base[S]**	0.2474	0.5382	0.1644	0.5000	0.8737	0.0539	NaN
5	**Base[+]**	0.2500	0.5000	0.1429	1.0000	0.0000	0.0000	NaN
6	**Base[-]**	0.0000	0.5000	0.0000	0.0000	0.8571	0.0000	NaN

3.2 Statistical Test

The Cochran's Q test returned the p-value 1.1395e-41. The following is the result of the pairwise McNemar:

Table 2: P-Values for Model Comparisons by McNemar

Model 1 Model 2	RF	LR	KNN	ADA	Base[S]	Base[+]
LR	0.0001					
KNN	0.8830	0.0003				
ADA	0.1496	0.0000	0.4011			
Base[S]	0.8243	0.0006	0.6530	0.2354		
Base[+]	0.0000	0.0000	0.0000	0.0000	0.0000	
Base[-]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

4 Discussion

The test results show how the models generalize to new individuals in the same context. By not using cohort, puzzler, individual, round, and phase information, potentially useful features are excluded.

Each individual is observed for 4 rounds with 3 phases each. Rounds and phases are evenly distributed between the test and validation sets. However, cohort and puzzler are unevenly distributed since each individual belongs to only one cohort and one puzzler class.

To understand the implications of this, a one-way ANOVA could be utilized to analyze the effect of individual features on the output variable. Interaction effects could be investigated using two-, three-, or four-way ANOVA. Considering the potential effects of these features, we explore the results:

Since the Cochran's Q test returned a very low p-value, there must be some differences in performance between the models. This is the reason for the pairwise McNemar.

Since LR have p-value under the significant threshold $p = 0.0024$ on its McNemar test with all other models, it does not have the same performance as any other models. The same is the case with Base[+] and Base[-].

Since all the intern pairwise comparisons between RF, KNN, ADA and Base[S] are above the significant threshold, we have to reject the null-hypotesis that there is a difference in performance between them.

- The NPV of Base[+] is 0 because it always predicts 1.
- Both the precision and recall of Base[-] are 0 since it always predicts 0. Consequently, the F1-score for Base[-] is also 0.
- Apart from the two baseline models, LR has the lowest scores across all metrics, and since all the pairwise McNemar tests show significant differences. This indicates LR had the worst performance.
- KNN has the highest scores across all metrics, but the pairwise McNemar tests show no significant difference between RF, ADA, KNN, and Base[S]. Therefor there is not enough evidence to conclude that KNN had best performance.
- It is noteworthy that ADA and KNN have better NPVs than Base[-], despite the imbalance favoring Base[-].
- Looking at Base[+], it has better performance than RF on all metrics,
- Generally, KNN and ADA have better performance than Base[+].
- Comparing RF and Base[-], they perform differently on different performance metrics.
- The MCC scores indicate that RF and LR perform worse than random guessing, whereas KNN, ADA, and Base[S] perform better.

One possible explanation for LR performing poorly is it lack of complexity compared to especially RF and ADA.

Even though the McNemar test did not show any difference in performance between RF, KNN, ADA and Base[S], since the KNN had the highest performance on all metrics, it could indicate that it has better performance. To further investigate this and the other performance differences, bootstrapped p-values and confidence intervals could be utilized.

In regards to generalization, we addressed it by using leave-one-group-out cross-validation, to ensure the models were evaluated on data from individuals it was not trained on.

Furthermore, each time a model is trained, the entire pipeline is rerun, introducing new over-sampled data, which better reflecting performance on new data. However, we cannot determine how well the models would generalize to different datasets or other situations, such as situations where individuals are not in a puzzling experiment. We would expect the models to perform worse in such scenarios, but this effect is mitigated by not using the round, phase, individual, or cohort features.

Robustness to change data is indirectly measured in the cross-validation. To further investigate this, one could log the metrics for test results on each outer fold, and compare them, potentially by their standard deviations.

Expanding the feature set could improve the models. Including factors like age, gender, sleep duration, and various health conditions might enhance model performance. Additionally, collecting more observations could improve the dataset.

In the inner folds grid search, we optimize for F1-score. This puts a bias on this metric, because the hyperparameters have been optimized with regards to this, on not other scores. By choosing which score the inner grid-searches optimize for, one could possible change performances of the models.

5 Conclussion

The following conclusion are from the context of the specific dataset. The dataset had a limited amount of observations, which was a limiting factor.

It can be concluded that LR had the worst performance among the models on this dataset.

KNN, ADA, and Base[S] performed better than Base[+] and Base[-]. RF performed worse than Base[+] and Base[S], it performed similar to Base[-].

For further research, this paper indicates that KNN may have a better performance. Base[S] should be used as the baseline since it is the best overall performing baseline.

References

Sneha Das, Nicklas Leander Lund, Carlos Ramos González, and Line H Clemmensen. Emopaircompete - physiological signals dataset for emotion and frustration assessment under team and competitive behaviors. In *ICLR 2024 Workshop on Learning from Time Series For Health*, 2024. URL https://openreview.net/forum?id=BvgAzJX40Z.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011. URL http://arxiv.org/abs/1106.1813.

Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A. F. Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview . *Bioinformatics*, 16(5):412–424, 05 2000. ISSN 1367-4803. doi:10.1093/bioinformatics/16.5.412. URL https://doi.org/10.1093/bioinformatics/16.5.412.