Danmarks Tekniske Universitet

Project in Statistical Evaluation for Artificial Intelligence and Data (02445)

Evaluating AI models - Individual Report

Oscar Thorsted Svendsen (S224177)

June 21, 2024

1 Task 1 - Evaluating AI Models in Research Papers

1.1 Paper 1 - *"Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis"* [1]

The first paper evaluates the application of artificial neural networks (ANNs) for predicting bankruptcy. It presents a comprehensive framework for employing ANNs and they use cross-validation to compare their performance with traditional logistic regression. The evaluation methodology in this paper is appropriate because it employs a robust training process, including backpropagation with gradient descent and a nonlinear optimizer. Additionally, the use of five-fold cross-validation ensures the model's robustness while utilizing the data effectively.

Moreover, the paper uses multiple evaluation metrics such as mean squared error (MSE), classification rates, and statistical significance (p-values), providing a thorough assessment of the model's performance. The incorporation of Bayesian classification theory also adds a solid theoretical foundation for the results.

Furthermore, the paper does a great job evaluating the AI models by conducting a comprehensive comparative analysis between ANNs and logistic regression . They use multiple statistical tests to check how reliable their results are. By incorporating techniques such as the receiver operating characteristic (ROC) curve analysis and comparing the area under the curve (AUC) values, the study provides a nuanced understanding of each model's predictive capabilities. This thorough evaluation helps highlight the strengths and weaknesses of each model and shows the importance of using different methods to fully understand how well the models work.

Lastly, the data is well documented and the paper makes sure to note that they didn't know exactly how the ANN made its decisions which poses a potential problem if it were to be implemented.

1.2 Paper 2 - *"Hospital quality classification based on quality indicator data during the COVID-19 pandemic"* [2]

The second paper aims to classify hospital quality using various machine learning algorithms based on quality indicator data collected during the COVID-19 pandemic. However, the evaluation methodology in this paper is much less rigorous compared to the first.

The study relies heavily on accuracy score as the sole evaluation metric, which can be misleading, especially in the context of imbalanced datasets common in healthcare. It does not provide a detailed comparison of how the different machine learning models impact the analysis or conclusions, which weakens the robustness of the findings. Additionally, the study does not adequately address potential biases in the data, such as those resulting from the uneven impact of the COVID-19 pandemic on different hospitals.

The interpretation of results focuses mainly on the model's performance but lacks depth in discussing the implications of data quality and completeness. To improve the evaluation, the paper should include a broader range of metrics such as precision, recall, F1-score, and AUC-ROC, and address data biases more thoroughly. These scores are mentioned briefly but not utilized adequately in the evaluation of the models. A detailed comparative analysis of multiple models and strategies for handling imbalanced data would also enhance the study's robustness. They could have used ANOVA on the different model metrics - even just on accuracy, but they only seem to compare which number is the largest.

They also did not calculate an estimate for their required sample size, nor did they do any considerations for class balancing. They briefly mention class balance in the evaluation along with the possibility of cross-validation - however, without mentioning Group K Fold which would be crucial. The report would have benefited greatly from having these aspects introduced during data preprocessing.

2 Task 2 - Building and Evaluating AI Models

2.1 Preparing the Data

2.1.1 Defining Model Parameters

The objective of this task was to build and evaluate at least two predictive models to classify the level of frustration from heart rate (HR) signals. The dataset, a subset of the EmoPairCompete dataset, included several HR features such as mean, median, standard deviation, minimum, maximum, and area under the curve (AUC), along with metadata on the rounds, phases, individual indices, and frustration levels.

As the objective of this study was to predict frustration from heart rate (HR) data, we only used those parameters in our models. We created the correlation heatmap in Figure 1 to test for the correlation between the HR parameters to see if any of them could be particularly good at predicting frustration. Or if there were any that we could reasonably leave out (this was done with a binarized frustration variable explained below).

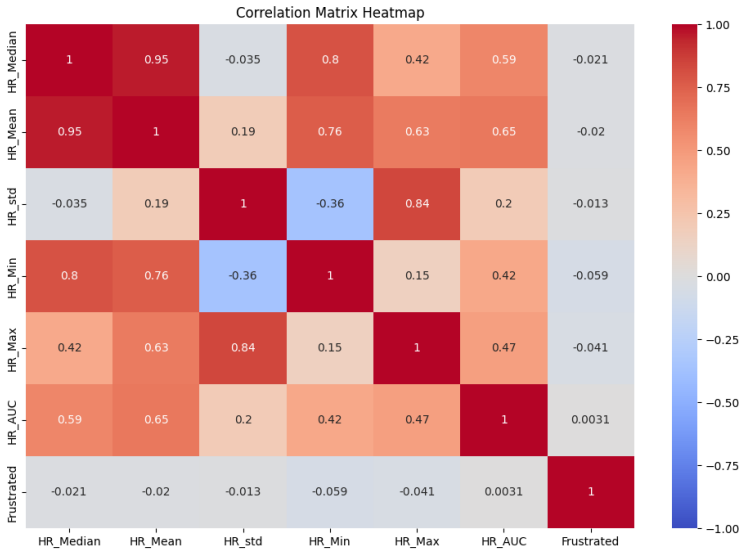


Figure 1: Correlation heatmap of model parameters

It became immediately obvious that there was very little correlation between frustration and heart rate data. Additionally, when comparing the HR parameters to each other, we observed that they all were within the same order of magnitude except "HR_AUC" which is had a correlation a factor 10 smaller. A case could be made to exclude this variable, but since the models in this report didn't require too much computing power, there was no apparent reason to exclude it - and when running the models later on, it did seem to improve accuracy just a bit. The same applied to the mean and median parameters which have a 95% correlation to each other, meaning that we could reasonably exclude one or the other without losing much information. But again, there was no obvious reason to do so, and when attempting it later in the process it resulted in worse accuracies.

We begun the pre-processing by binarizing the frustration levels. The frustration variable was initially continuous, but due to the small size of the dataset, it was essential to convert this into a binary format. To determine an appropriate binarization threshold, we considered different values. A threshold that was too low, such as the median value of 2, might not adequately represent a significant frustration level, making it problematic to justify classifying someone as frustrated at such a low threshold. On the other hand, a higher threshold increased the risk of class imbalance, which could negatively affect model training and evaluation. To test this, we first had to flesh out the cross-validation strategy.

2.1.2 Cross-Validation

For evaluating our models, we employed cross-validation to ensure robust performance metrics and to make the best use of the limited data. Given presence of groups in the dataset consisting of datapoints from the same individuals, we used Group-K-Fold. Group-K-Fold ensured that the data from the same individual did not appear in both training and test sets, which helped generalize the model's performance to unseen individuals.

In addition to groups, we were also working with classes in our data. To ensure class and group balance, we would usually apply Stratified-Group-K-Fold. However, although Stratified-Group-K-Fold could potentially help with class imbalance, we suspected that our dataset might be too small for both groupings to have an effect. We conducted several tests where we measured the class balance between the different folds and between the train and test data to analyze the impact of the two cross-validation strategies. We found that stratification did not improve the class balance compared to Group-K-Fold. Therefore, the rest of the study is conducted using Group-K-Fold.

We then experimented with different numbers of folds and binarization thresholds to see how they affected class balance and model performance. Our findings indicated that a binarization threshold of 5 and above resulted in class imbalances, such as some folds having no frustrated individuals in the test set, leading to non-representative performance metrics, non-normally distributed accuracies, and inhomogeneous variances. All of which we would need to perform ANOVA later in the paper. We, therefore, settled on a threshold value of 4. This threshold provided a reasonable balance between a meaningful frustration level and maintaining class balance.

Lastly, we tested the the highest number of folds that maintained decent class balance with a binarization threshold of 4. In our attempt to get as much out of our limited dataset as possible, we needed to aim for as many folds as possible with a natural maximum at 14 due to the dataset containing 14 individuals in our Group-K-Fold. The highest number of folds that maintained class balance turned out to be 12.

3 Training the Models

We selected Logistic Regression and Random Forest Classifier as our primary machine learning models for several reasons. Logistic Regression is a widely used algorithm for binary classification problems due to its simplicity, interpretability, and efficiency. It estimates the probability that a given input point belongs to a certain class and works well when the relationship between the features and the target variable is approximately linear. Given the nature of our binary classification task, Logistic Regression provided a straightforward approach to modeling this relationship.

Random Forest, on the other hand, is a versatile and powerful ensemble learning method. It builds multiple decision trees and merges them to get a more accurate and stable prediction. Random Forest is often useful because it can handle a large number of input features, and manage missing data well through its inherent bootstrap sampling and feature randomness. These abilities were not particularly important in our case, so the model was mostly included to observe how a more complex model would handle the data.

Both models have their drawbacks. Logistic Regression assumes a linear relationship between the independent variables and the dependent variable, which might not be the case with our data. It is also sensitive to outliers and can perform poorly if the data is not properly scaled. Random Forest, while robust and versatile, can be computationally expensive, especially with a large number of trees and high-dimensional data - this was not an issue in this paper due to the limitations of the data. It also tends to be less interpretable than simpler models like Logistic Regression, as it is difficult to understand the contribution of each individual feature to the final prediction.

For baseline model comparison, we used a Dummy Classifier that simply made predictions based on the most frequent class.

We defined our models in a dictionary for easy reference and iterated over them to perform cross-validation using *cross_validate* from *sklearn*. For each model we stored the train and test accuracy of all 12 folds which would later be used to compare the models using ANOVA and various robustness, generalization, and consistency tests.

Boxplots of the test accuracies can be seen in Figure 2. The first thing that we noticed was that the Random Forest model exhibited much less variance in accuracies than the two other models. This could be

a sign of a more robust model, which will be analyzed later in the report.

Secondly, we could deduce that the none of the two trained models seemed to greatly outperform the Baseline model (which will be tested with ANOVA) and that Logistic Regression and the Baseline model had extremely varying accuracies across our folds. This could be due to various factors such as class imbalance, a small sample size, and variability in the individual data. The class imbalance problem was, unfortunately, amplified by using Group-K-Fold, since balancing it for individuals might imbalance it for classes. There was not much that we can do that was not already done to prevent these issues.

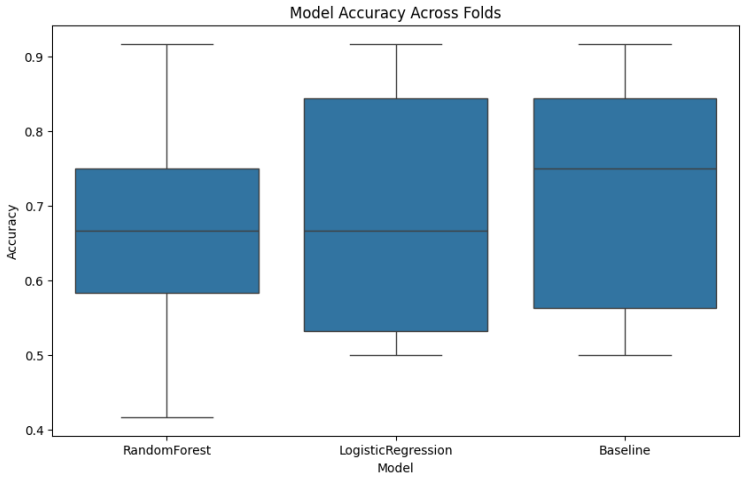


Figure 2: Boxplots of accuracies in each CV fold of the 3 models

4 Model Testing

4.0.1 ANOVA

To test whether there was a significant difference between the accuracies of the models, we performed a one way ANOVA test using *f_oneway* from *scipy.stats*. However, before doing so, we checked for the following ANOVA assumptions:

Normally distributed observations within each group
In this scenario, the observations within each group were the test accuracies for each fold. To test for a normal distribution, we performed a Shapiro-Wilk test on each model - once again using *scipy.stats*. In addition, we plotted Q-Q plots for for the accuracy scores. Both the plots and p-values can be seen in Figure 3. The p-values all indicate normality, and while the Q-Q plot for Logistic Regression might not seem perfectly normal, we could still conclude that the test accuracies for all models were normally distributed to a statistically significant degree.

Homogeneous variances
Similar to the test above, we performed a Levene's test from *scipy.stats* to test whether the variances of the test accuracies were homogeneous. With a p-value of 0.98, we could conclude that the variances indeed were equal, which satisfied the requirement for ANOVA.

Independent residuals with 0 means
Lastly, we checked the means and independence of the residuals using a Durbin-Watson test. The residuals were calculated as the difference between the group means and the observed test accuracies. As can be seen in Figure 4, the means were all very close to 0 (which was a given, due to the definition of the residuals in this case). The Durbin-Watson statistic was close to 2 in all models, which signified independent residuals.

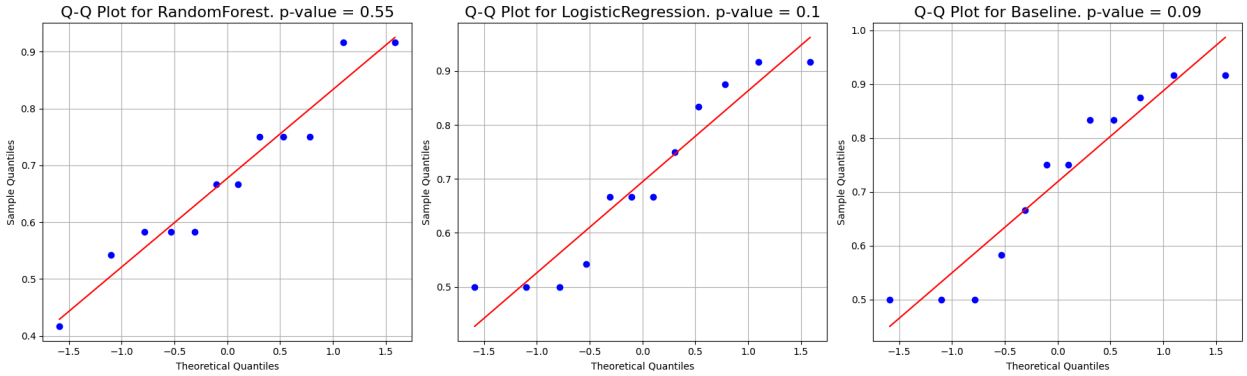


Figure 3: Q-Q Plots of test accuracies for the 3 models

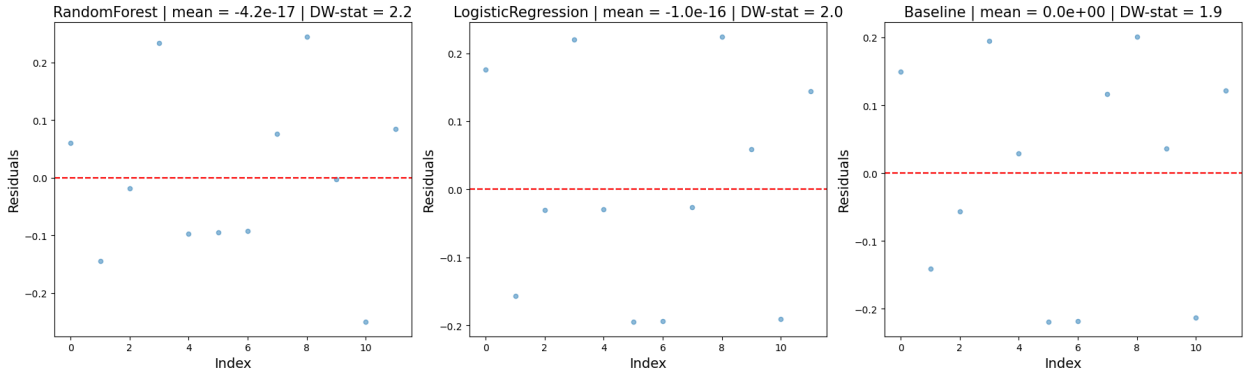


Figure 4: Scatterplots of residuals for the 3 models including mean and Durbin-Watson statistic

With the assumptions for ANOVA checked, we could move on to the actual ANOVA test. With the null hypothesis that all the models had identical test accuracies, we ended with a p-value of 0.81. We could therefore not reject the hypothesis that all 3 of the models performed equally well (including the baseline). The mean accuracies of the models can be found in Table 1 and it is clear that none of the models perform extraordinarily well in predicting frustration.

Models	Baseline	Random Forest	Logistic Regression
Mean accuracy	71.9%	67.7%	69.4%

Table 1: Mean accuracies for the baseline and the 2 trained models

4.1 Further Comparison of the Models

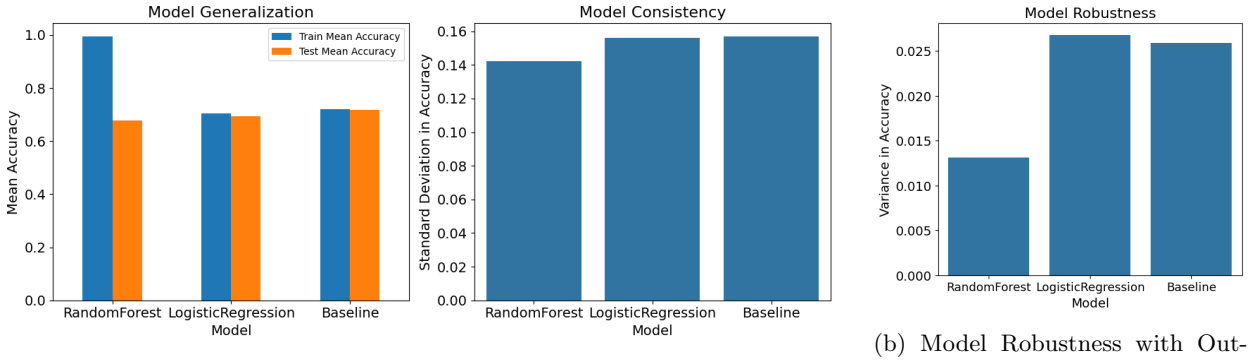
Now that we had concluded that the two trained models perform just as well as the baseline and each other, we tested them for generalization, consistency, and robustness to detect whether any of them could be preferable to the other.

The first tests were on generalization and consistency. To test for generalization, we examined the mean train and test accuracies of each model, and for consistency, we plotted the standard deviation in test accuracy. From Figure 5a we could see that Random Forest showed a significant drop in accuracy from train- to test data, indicating that it may have be overfitting to the training data and not generalizing well to unseen data. This might have been preventable by hyperparameter tuning, regularization, feature selection etc. The Logistic Regression and Baseline models had smaller gaps between their training and test

accuracies, indicating better generalization.

Regarding consistency, all the models had similar standard deviations in test accuracy, with Random Forest having the lowest value, indicating slightly higher consistency compared to Logistic Regression and Baseline models.

Afterward, we tested model robustness to changes in data. Here, we manually introduced outliers in the dataset with extreme HR- or frustration values and retrained the models. The plotted variance in test accuracies can be seen in Figure 5b. The Random Forrest model seemed to have a significantly lower variance in accuracy, which indicated that it was more robust to outliers and changes in the data. This was expected as this more complex model is known for being more robust.



(a) Generalization and Consistency Plots

(b) Model Robustness with Outliers

Figure 5: Comparison of model performance metrics: Generalization, Consistency, and Robustness

4.2 Conclusion

In this study, we aimed to predict frustration levels from heart rate signals using machine learning models. We selected Logistic Regression and Random Forest Classifier as our primary models, and employed a Dummy Classifier as a baseline for comparison. Through extensive data preprocessing, including the binarization of frustration levels and the use of Group-K-Fold cross-validation, we ensured robust evaluation metrics.

Our analysis showed that Logistic Regression and Random Forest both performed similarly to the baseline model in terms of accuracy, as confirmed by an ANOVA test. Despite the Random Forest model exhibiting higher robustness to data variability and outliers, it showed signs of overfitting, indicated by the significant drop in accuracy from training to test data. Logistic Regression, while simpler, provided better generalization with consistent performance across folds.

The study highlights the challenges of working with small datasets and the importance of choosing appropriate evaluation strategies. Future work could focus on collecting more data and exploring advanced models or hybrid approaches to improve prediction accuracy and model robustness further.

5 References

References

[1] Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. https://www.sciencedirect.com/science/article/pii/S0377221798000514, 1999.

[2] Nurhaida Ida. Dhamanti Inge. Ayumi Vina et al. Hospital quality classification based on quality indicator data during the covid-19 pandemic. https://doi.org/10.11591/ijece.v14i4.pp4365-4375, Aug 2024.

See *"Appendix 1"* for the code for the project.