**Introduction**

The electrocardiogram (ECG) is essential, for monitoring and detecting heart conditions such as arrhythmias, ventricular fibrillation and blockages without procedures. It's crucial to identify heartbeats like contractions (PVCs) and atrial fibrillation early on to help in effective treatment and prevent heart disease and Machine learning algorithms have gained popularity in categorizing ECG heartbeats due to their ability to analyze data efficiently and provide insights and algorithms like k neighbors (KNN) logistic regression and convolutional neural networks (CNN) are commonly used in ECG analysis. KNN is valued for its simplicity in identifying neighboring data points for classification and Logistic regression involves using a function to analyze variables, like distinguishing between heartbeat types or their absence and On the hand CNN uses layers to extract features from ECG signals and group them accordingly and this study aims to evaluate the performance of KNN, regression and CNN models in classifying ECG heartbeats and in this study utilize a dataset that includes ECG signals to train and evaluate these models for our analysis and the assessment of each models performance will consider metrics such, as accuracy, precision, recall and F1 score (Escalona Moran et al., 2015).The results of this study will provide information on the strengths and weaknesses of each model helping to guide the selection of ECG heartbeat classification methods.

The data provided offers an overview of a dataset called combined\_df that has been standardized. This dataset comprises 188 characteristics (columns) and 109,446 data points (rows). The summary consists of the count, deviation, minimum 25th percentile, median (50th percentile) 75th percentile and maximum values, for each feature.

The count plot here shows how the different classes are distributed in the combined\_df dataset. On the axis we have the classes while on the vertical axis we see the number of occurrences, for each class and the plot highlights an imbalance as class 0 has the count (around 80,000) and class 4 has the lowest count (about 20,000). This imbalance could affect how well machine learning models perform when trained on this dataset and fails to calucate the accuracy, f1-score, recall and precision.

The provided code creates two representations; Class Distribution Before Balancing and Class Distribution After Balancing. Both visuals illustrate the frequency of classes in a dataset.

Class Distribution Before Balancing; The initial visualization depicts the distribution of classes before any adjustments with counts varying between 2500 and 80000. This graph highlights an imbalance, in class representation, where one class dominates over the others.

Class Distribution After Balancing; In contrast the second visualization showcases the outcomes balancing efforts. Displaying counts ranging from 0 to 4.0 this graph demonstrates an distribution among the classes after implementing balancing techniques. This shift indicates an enhancement in achieving balance.

Key Contrasts:

The primary disparity between the two visuals lies in the degree of class imbalance observed. While the first plot exhibits disparities the second one illustrates an even distribution following balancing interventions. This implies that methods like oversampling or undersampling were utilized to address class imbalances within the dataset. This process has led to an equilibrium, among classes potentially enhancing machine learning model performance.

**Methodology**

**K Nearest Neighbors (KNN)**

The KNN method is a type of learning technique that is employed in tasks related to classification and regression. It operates by identifying the 'k' most instances (, nearest neighbors) to an instance and then utilizing their labels to forecast the label of the new instance. This approach though straightforward proves to be effective. Is applicable, for both multi class classification issues.(Ahamed et al., 2020)

**Logistic Regression**

Logistic Regression, another learning method is utilized for addressing classification challenges. It gauges the likelihood of an instance being part of one of two classes based on a series of predictor variables. Through utilizing a function to depict the probability of the response variable this technique can be employed in scenarios involving both multi class classification problems. (Escalona-Moran et al., 2015)

When comparing the training sessions of Logistic Regression and KNeighborsClassifier we looked at the accuracy and classification reports. The Logistic Regression model had an accuracy rate of 76.90% while KNeighborsClassifier achieved an accuracy of 96.88% and the classification reports indicated that KNeighborsClassifier performed better than Logistic Regression in terms of precision, recall and f1 score across all classes and these findings suggest that KNeighborsClassifier might be more appropriate, for this dataset and task. However, analysis and validation such, as hyperparameter traning and validation accuracy and validation loss testing on a dataset are needed to confirm the models effectiveness.

A Convolutional Neural Network (CNN) is a learning model created to analyze visual data and It comprises layers that carry out functions, like convolution, pooling and activation to extract characteristics from images. These characteristics are then utilized for tasks like classification, detection or other purposes. CNNs are extensively utilized in computer vision applications such as image recognition, object detection and facial recognition because of their capability to automatically grasp representations of features and due to their hierarchy and shared weights CNNs effectively capture relationships in data making them formidable tools, for tasks that involve understanding and processing visual information.(Issa et al., 2023)

The CNN model has demonstrated an accuracy of 98.35%, in predicting the desired outcomes. The classification report provides details on precision, recall and f1 score for each class reflecting how well the model performs. For example a precision score of 0.98 for class 0.0 means the model is 98% of the time when predicting that class, Similarly a recall score of 0.95 for class 0.0 indicates the model correctly identifies that class 95% of the time and The f1 score, which combines precision and recall is 0.97 for class 0.0 with performance seen across classes as well and both macro average and weighted average scores are high indicating performance, across all classes. Overall accuracy stands at 98% showing the models proficiency in assigning classes to input data.

**Hyperparameter Tuning**

Fine tuning the parameters, for K Nearest Neighbors (KNN) involves adjusting factors such as the number of neighbors, distance metric and weights, to find the combination of these parameters methods like grid search or random search can be utilized and for instance in scikit learns KNeighborsClassifier you can tweak parameters such as the number of neighbors, weights and distance metric using GridSearchCV. (Yunendah Nur Fuadah, Muhammad Adnan Pramudito and Ki Moo Lim, 2022)

When it comes to Logistic Regression key hyperparameters like regularization strength (C) and solver can be adjusted, in scikit learns LogisticRegression module you have the option to fine tune parameters such as regularization strength and solver type and penalty using GridSearchCV.(El-Hasnony et al., 2022)

In the case of a Convolutional Neural Network (CNN) tuning hyperparameters like learning rate, batch size and layer count is crucial, in platforms like TensorFlow or Keras parameters such as number of filters, kernel size, activation function choice and optimizer selection can be optimized through techniques like GridSearchCV or Bayesian Optimization and it is worth noting that hyperparameter tuning for CNNs can be resource intensive and time consuming due to their nature. (Maghawry et al., 2021)

Before adjusting the hyperparameters the KNeighborsClassifier model showed performance achieving an accuracy of 0.9688 and delivering f1 scores ranging from 0.94, to 0.99 across all classes. However post hyperparameter tuning the models accuracy notably rose to 0.9816. The f1 scores for all classes saw enhancements, now varying between 0.96 and 0.99 while the recall for class 0 also improved from 0.91 to 0.93 and this suggests that tweaking the hyperparameters played a role in boosting the models performance across all classes. In summary both the macro and weighted f1 scores increased from 0.97 to 0.98 post tuning signaling an enhancement, in the models performance.

The Logistic Regression model showed performance initially with an accuracy of 0.7690 before adjusting the hyperparameters. It excelled in categories. Lagged behind in others. Following hyperparameter tuning the accuracy saw an uptick to 0.7734. While some categories saw improvement others experienced a decline, in performance. Despite these fluctuations the overall macro and weighted f1 scores remained steady at 0.77 indicating that the models overall effectiveness did not undergo alterations.

The CNN model showed performance without adjusting hyperparameters achieving an accuracy of 0.9835 and f1 scores ranging from 0.97, to 1.00 across all classes. Following hyperparameter tuning improvements were seen in class performance. The f1 scores for classes 0 1 and 2 rose to 0.98, 0.98 and 0.99 respectively with class 2s recall increasing from 0.98 to a score of 1.

While the CNN models accuracy remained tuning enhancements in f1 scores and recall for select classes led to an uptick in both macro and weighted average f1 scores from 0.98 to 0.99. In general the CNN model delivered results pre and post hyperparameter tuning with the showcasing improved performance, in specific class outcomes.

In the graph you can observe the training and validation loss trends of the KNN model and the training loss decreases steadily over time indicating that the model is effectively learning from the data and on the hand while the validation loss also decreases and this graph illustrates how well the KNN model performs during training and validation across epochs and during training the accuracy consistently remains high at around 0.98 whereas, the validation accuracy fluctuates between 0.96 and 0.88.(Huang, Li and Yu, 2021)

Comparison of CNN Training and Validation Accuracy and Loss

Based on the plots provided and can observe the following trends:

Accuracy

The accuracy plots for both training and validation, in the CNN model exhibit a pattern showing an increase as the number of epochs progresses and the training accuracy peaks at 0.98, while the validation accuracy peaks around 0.96 and this indicates that the model effectively captures patterns in the training data although there is an issue of overfitting as the validation accuracy lags behind the training accuracy and training loss. (Huang, Li and Yu, 2021)

Loss

Regarding loss both training and validation loss curves in the CNN model follow a trend of decreasing as epochs increase. and the training loss experiences decline initially before stabilizing at around 0.15 and on the hand the validation loss decreases at a pace and levels off near 0.20 and this implies that while the model successfully minimizes loss on training data there exists a disparity. Between training and validation losses hinting at some degree of overfitting and traning accuracy and traning loss. (El-Hasnony et al., 2022)

After analyzing the ROC and PR curve results it's clear that the K Nearest Neighbors (KNN) Logistic Regression and Convolutional Neural Network (CNN) models performance, in classification is showcased and these curves illustrate how the True Positive Rate (TPR) and False Positive Rate (FPR) trade off at threshold values and the Area Under the ROC Curve (AUC) is used to assess the models where an AUC of 1 signifies a classifier and 0.5 represents a random classifier.(Seyed Matin Malakouti, 2023)

The AUC scores for each model further confirm their performance and the CNN model particularly shines with AUC scores of 1.00 across all classes indicating performance consistency and additionally both the KNN and Logistic Regression models exhibit performances with AUC values approaching 1.00. (Seyed Matin Malakouti, 2023)

**Comprehensive Evaluation**

The comprehensive examination and comparison of K Nearest Neighbors (KNN), Logistic Regression, and Convolutional Neural Network (CNN) models for categorizing ECG heartbeats is truly remarkable and the detailed analysis of performance metrics such as accuracy, precision, recall, and F1 score provides valuable insights into the strengths and weaknesses of each model and the visual representations of class distributions before and after balancing the dataset effectively highlight the significance of addressing class imbalances and the methodology section clearly elucidates how each model approaches the classification of ECG heartbeats.

The results obtained from hyperparameter tuning for KNN and Logistic Regression models indicate the potential for improving model performance through parameter adjustments and furthermore the post-tuning accuracy and F1 scores of the CNN model demonstrate the substantial impact of this process on enhancing performance and the analysis of ROC and PR curves and along with AUC scores and serves to validate the models performance and underscoring their reliability in classification tasks in these models.

Further discussion on the real-world implications of these findings and an exploration of the resources required for training and tuning the CNN model would enhance the practical understanding of deploying these models.

Overall, the thorough examination, comprehensive comparisons, and emphasis on performance assessment significantly contribute to the study of ECG heartbeat classification.

**Conclusion**

In conclusion provides valuable insights into the strengths and weaknesses of K Nearest Neighbors (KNN), Logistic Regression, and Convolutional Neural Network (CNN) models for classifying ECG heartbeats. The thorough examination of performance metrics, visualization of class distribution, and the impact of hyperparameter tuning emphasize the significance of these models in heart condition detection. Additionally, the analysis of ROC and PR curves alongside AUC scores validates the reliability of the models in classification tasks and further exploration of real-world implications and resource requirements for practical deployment would enhance the study's practical relevance and value.

**References**

Ahamed, Md.A., Hasan, K.A., Monowar, K.F., Mashnoor, N. and Hossain, Md.A. (2020). ECG Heartbeat Classification Using Ensemble of Efficient Machine Learning Approaches on Imbalanced Datasets. *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*. doi:https://doi.org/10.1109/icaict51780.2020.9333534.

Ahmad, Z., Tabassum, A., Guan, L. and Khan, N.M. (2021). ECG Heartbeat Classification Using Multimodal Fusion. *IEEE Access*, 9, pp.100615–100626. doi:https://doi.org/10.1109/access.2021.3097614.

Alarsan, F.I. and Younes, M. (2019). Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. *Journal of Big Data*, 6(1). doi:https://doi.org/10.1186/s40537-019-0244-x.

El-Hasnony, I.M., Elzeki, O.M., Alshehri, A. and Salem, H. (2022). Multi-Label Active Learning-Based Machine Learning Model for Heart Disease Prediction. *Sensors*, 22(3), p.1184. doi:https://doi.org/10.3390/s22031184.

Escalona-Moran, M.A., Soriano, M.C., Fischer, I. and Mirasso, C.R. (2015). Electrocardiogram Classification Using Reservoir Computing With Logistic Regression. *IEEE Journal of Biomedical and Health Informatics*, 19(3), pp.892–898. doi:https://doi.org/10.1109/jbhi.2014.2332001.

Huang, Y., Li, H. and Yu, X. (2021). Number 6 Citation Youhe Huang et al. *Physics and Engineering in Medicine Physiological Measurement*, 42, p.65003. doi:https://doi.org/10.1088/1361.

Issa, M.F., Yousry, A., Tuboly, G., Juhasz, Z., AbuEl-Atta, A.H. and Selim, M.M. (2023). Heartbeat classification based on single lead-II ECG using deep learning. *Heliyon*, 9(7), p.e17974. doi:https://doi.org/10.1016/j.heliyon.2023.e17974.

Maghawry, E., Gharib, T.F., Ismail, R. and Zaki, M.J. (2021). An Efficient Heartbeats Classifier Based on Optimizing Convolutional Neural Network Model. *IEEE Access*, 9, pp.153266–153275. doi:https://doi.org/10.1109/access.2021.3128134.

Oyeleye, M., Chen, T., Titarenko, S. and Antoniou, G. (2022). A Predictive Analysis of Heart Rates Using Machine Learning Techniques. *International Journal of Environmental Research and Public Health*, 19(4), p.2417. doi:https://doi.org/10.3390/ijerph19042417.

Seyed Matin Malakouti (2023). Heart disease classification based on ECG using machine learning models. *Biomedical Signal Processing and Control*, [online] 84, p.104796. doi:https://doi.org/10.1016/j.bspc.2023.104796.

Silva, P., Luz, E., Silva, G., Moreira, G., Wanner, E., Vidal, F. and Menotti, D. (2020). Towards better heartbeat segmentation with deep learning classification. *Scientific Reports*, 10(1). doi:https://doi.org/10.1038/s41598-020-77745-0.

Yunendah Nur Fuadah, Muhammad Adnan Pramudito and Ki Moo Lim (2022). An Optimal Approach for Heart Sound Classification Using Grid Search in Hyperparameter Optimization of Machine Learning. 10(1), pp.45–45. doi:https://doi.org/10.3390/bioengineering10010045.