**Week 9 Neural Network for Tabular Data**

**Milestones**

* Train a perceptron model one versus rest classifier for entire data and selected featured data.
* Visualize decision boundaries for our features with binary classes, then multi class across both the data sets.
* Train a multiclass classification neural network and perform cross-validation in order to evaluate model overfitting.
* Model evaluation based on several aspects was carried out, that are, accuracy, precision, recall, F1-score, and AUC.
* Conducted hyperparameter tuning where learning rate, the number of layers and activation functions were varied in order to optimize the design.
* Implemented and compared different MLP classifier architectures and selected one that achieved a balanced score for accuracy and AUC scores.
* Studied the effect on the performance metrics of the model with the best architecture on different Classifiers with different splits.
* Detect the overfitting of the model via investigation of the variance of the prediction accuracy on the training data and on the testing data.
* Applied Early Stopping and dropout during training of the network to evaluate how it can balance overfitting.

**Summary**

* Cross validation was used to enhance the performance of a constructed neural network model dedicated to multiclass classification. Many classes performed well in the application of the model, although some classes exhibited differences across the board.
* The best architecture was chosen based on maximum accuracy and AUC scores of the model after hyperparameter tuning which also improved performance.
* Performance of the model was performed using different datasets, which were partitioned in different splits for training and testing of the model respectively. The critical variables showed that although accuracy and precision were in the range of above 90%, recall and F1 scores were class dependent. The confusion matrix made it possible to gain deeper understanding into all records that were wrongly classified.
* Overfitting was diagnosed based on the observation of the difference in accuracy between the training and test datasets.

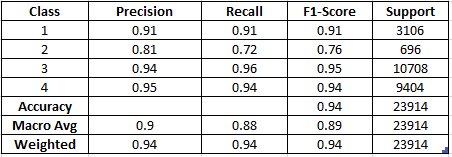
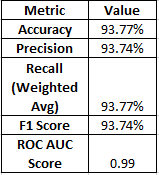
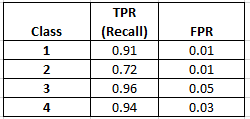
**Conclusions**

* Cross-validation enhanced the strength of the model and areas of its practical applicability by stressing the areas with class imbalances which were not dealt with effectively including in cross-validation.
* Neither the complete set nor the feature selected model would however fail the model in terms of performance since both obtained relatively high accuracy, hence the model’s perspectives to a greater extent are quite robust.
* Hyperparameters tuning improved the overall model performance, with the best accuracy (95%) and AUC achieved using the Adam Optimizer on deeper (256 × 256 depth) models. Sensitivity and recall were always constant and high while the specificity was variable between models. (for entire data)
* As it stands, architecture layer 6 ReLU and Adam optimizer provided the best results consistently. There was also general increase in accuracy and AUC with increase in layer size and using Adam optimization with SGD performing well but for accuracy lower values were observed. (for feature selected data)
* The model does have a reasonable performance as there is a sharper decline in both train accuracy and test accuracy after 60% of the data is shifted. The fact that the two measures, training & test accuracies, are reasonably close suggests there is no overfitting of the model.

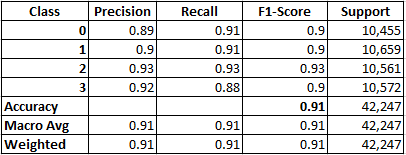
**Conclusions with tables and plots from Experiments**

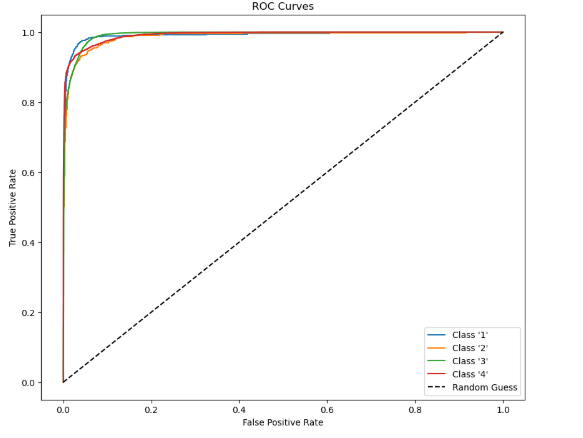
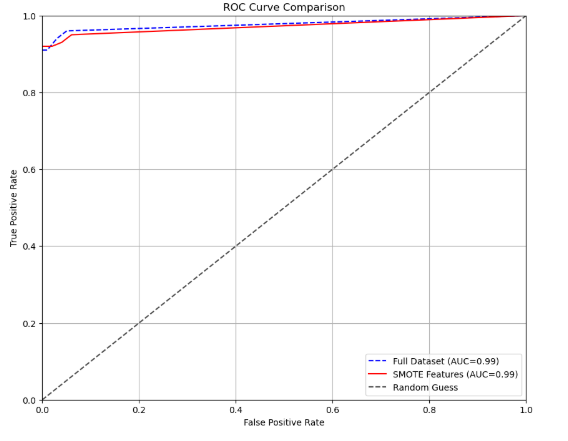
**Multiclass Classification with Neural Network, Cross-Validation, and Performance Metrics Evaluation**

For entire data

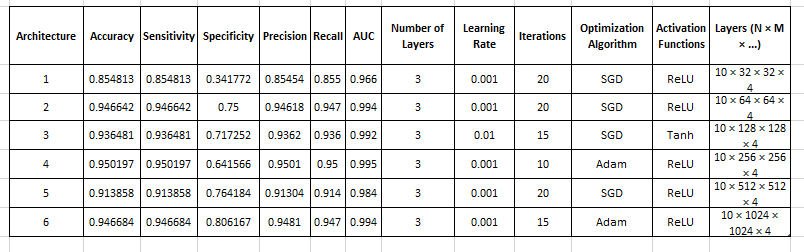
for selected features



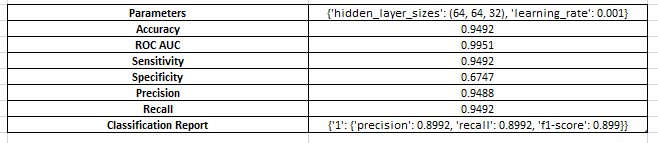
 

**Hyperparameter Tuning and Model Evaluation for Neural Network Classification**

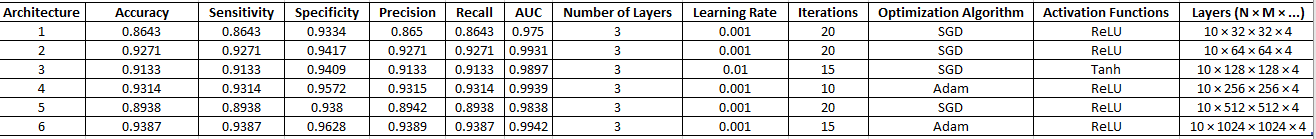
**Results for data set**

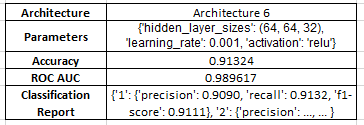


**Architecture 6 is the best**



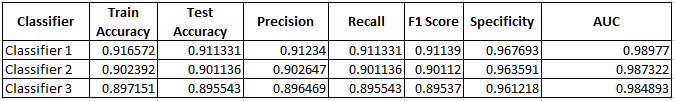
**For Selected Features**



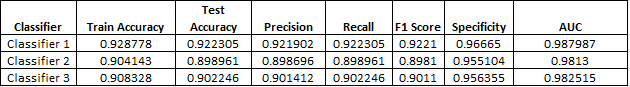


**Performance Evaluation of MLP Classifiers with Varying Training Data Splits: Accuracy, Precision, Recall, F1 Score, and Confusion Matrix**

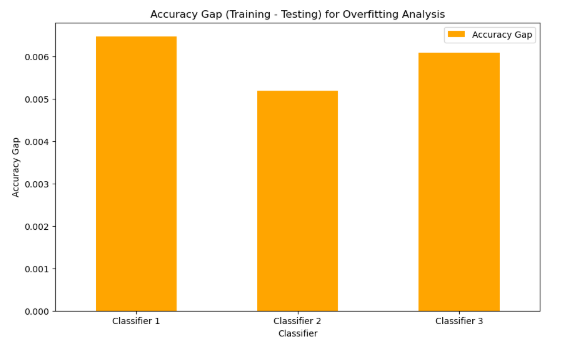
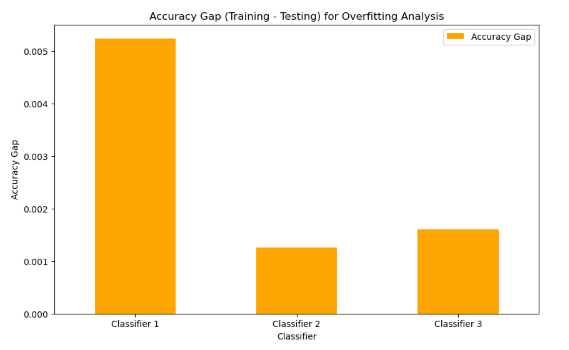
**For data set**



**For selected features**



**Overfitting Analysis: Identifying Accuracy Gaps Between Train and Test Data for MLP Classifiers**

**References**

Ageron, A. (2024). Neural networks with Keras. In *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). Retrieved from <https://colab.research.google.com/github/ageron/handson-ml3/blob/main/10_neural_nets_with_keras.ipynb>

GeeksforGeeks. (2024). sklearn model hyper-parameters tuning. [Online]. Available at: <https://www.geeksforgeeks.org/sklearn-model-hyper-parameters-tuning/> [Accessed 15 November 2024].

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hirok4. (2019). Building a multi-layer perceptron from scratch. *Medium*. Available at: <https://medium.com/@hirok4/building-a-multi-layer-perceptron-from-scratch-c9679752cf48> [Accessed 21 November 2024].

Heriot-Watt University. (2024). Neural Network Colab Notebook. Available at: <https://colab.research.google.com/drive/156DoGatav3nLxjY9sKeuGb8L5FE5KPgY?usp=sharing> (Accessed: 16 November 2024).

Scikit-learn Documentation. (n.d.). Model Selection: Cross-Validation and Metrics. Retrieved from <https://scikit-learn.org/stable/modules/cross_validation.html>

TensorFlow Tutorials. (n.d.). Neural Networks and Regularization Techniques. Retrieved from <https://www.tensorflow.org/tutorials>