****

**Name: Muhammad Junaid Fida**

**Registration No: 04072113048**

**Course Code: CS – 414 – AI**

**Semester: BS 5th**

**Department: Computer Science**

**Submitted To: Dr. Ayaz Hussain**

**A Report on Tic-Tac-Toe End-Game**

**(Binary Classification with ML Models: NN)**

**Problem/Dataset Description:**

We are classifying Tic-Tac-Toe game configurations as either winning ('positive') or non-winning ('negative'). We use Neural Network models to solve this binary classification problem. The dataset consists of Tic-Tac-Toe game configurations with features representing X's and O's in the 3x3 grid and the class label indicating whether the configuration leads to a winning outcome ('positive') or not ('negative'). The dataset is taken from Kaggle. It comprises 958 instances with nine features representing the board state and a binary class label indicating the positive or negative outcome.

**Model Configuration:**

We experiment using NN with varying numbers of hidden units and learning rates. Early stopping is employed to prevent overfitting using a validation set.

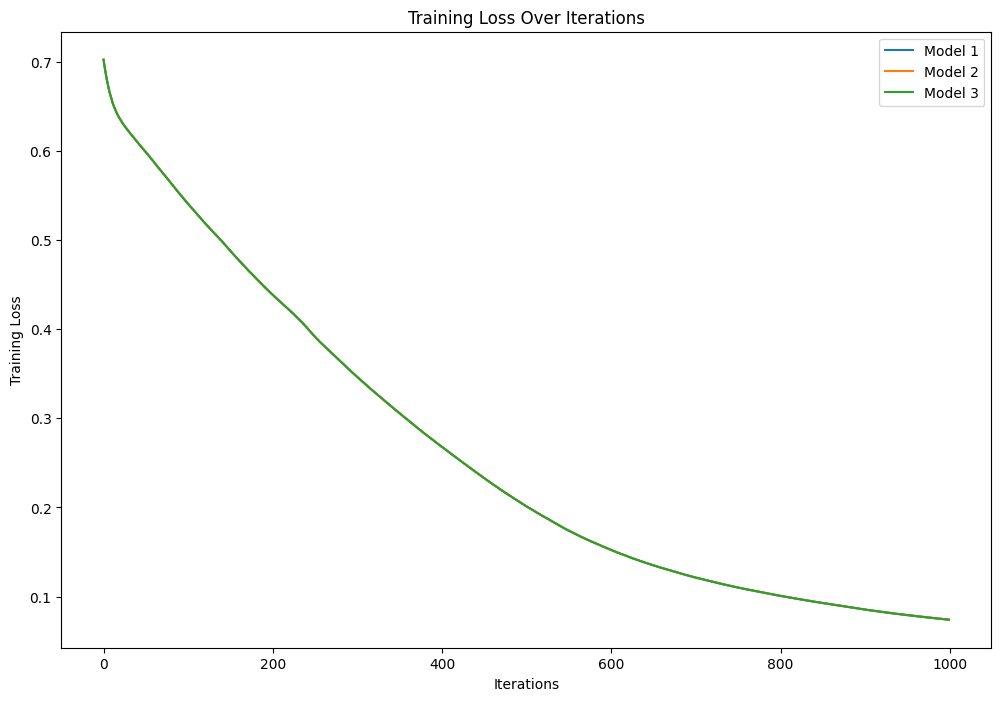
**1. Data Preprocessing:**

* The dataset was split into training, validation, and test sets to facilitate model training, tuning, and evaluation.
* Categorical features were one-hot encoded, and numerical features were scaled using `StandardScaler`.

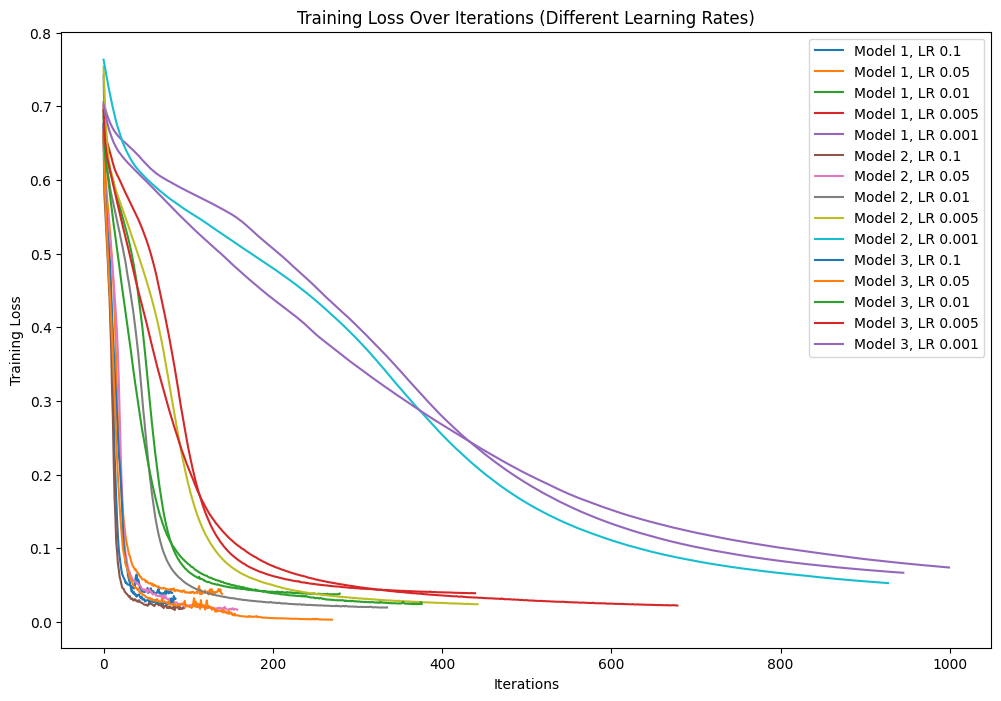
**2. Model Training:**

* The NN models were constructed with varying numbers of hidden units and learning rates.
* A range of hidden units was explored, from 2 to 5, to understand the impact of model complexity on performance.
* Different learning rates (0.1, 0.05, 0.01, 0.005, 0.001) were tested to find an optimal balance between training speed and convergence.

The below graph shows training loss over iterations for Neural Network Model

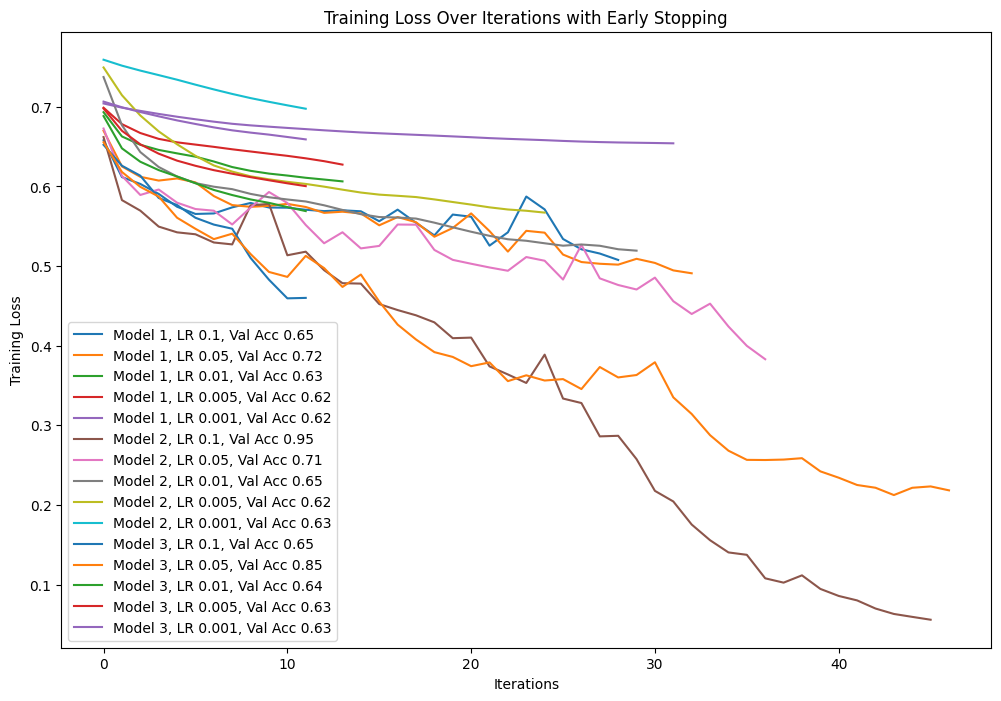


* This below graph shows learning curves of our models using different learning rates 0.1,0.05,0.01,0.005,0.001



**Early Stopping Criteria:**

* This graph demonstrates how your models are avoiding overfitting using early stopping. Validation Accuracy at different Learning Rates is also shown respectively in Graph for all 3 models
* Early stopping was implemented using the `early stopping` parameter in scikit-learn's MLPClassifier.
* The `validation\_fraction` parameter determined the proportion of the training data used for validation, and `n\_iter\_no\_change` set the number of epochs with no improvement on the validation set to trigger early stopping.
* To evaluate the model's performance during training, a validation set was created by splitting a portion of the training data (10%) aside. This validation set was not used for training; instead, it served as an independent dataset to monitor the model's performance.



**3. Results and Observations:**

* The training loss curves indicate that the models learned well without significant overfitting.
* The validation accuracies helped in selecting the best-performing model, demonstrating the effectiveness of early stopping.
* The chosen model achieved promising accuracy on the test set, validating its generalization to unseen data.

**Conclusion:**

The implemented MLP models, with careful tuning and early stopping, proved effective in classifying Tic-Tac-Toe game outcomes.

THE END!