Assessment Guidelines for **ILO 6.0: Modelling, Evaluation and Iteration**

This ILO consists of the following tasks:

- **Select a suitable machine learning problem:** Based on their proposal, they need to select a specific category of machine learning models (e.g., classification, regression, clustering, time series prediction, and so on).
- **Train a machine learning model:** Each team needs to try a relevant classical machine learning algorithms from the related category. The categories they may consider for their project are as follows:
    - *Classification*: Logistic regression, Decision Trees, Random Forest, Support Vector Machine (SVM), K-nearest neighbors (KNN), Gradient Boosting Machine, XGBoost, Multi-layer Perceptron classifier (using MLPClassifier from scikit-learn)
    - *Regression*: Linear regression, Ridge regression, Lasso regression, Elastic net, Polynomial regression, Support vector regression, Decision Tree regression, Random forest regression, Gradient boosting regression, XGBoost, K-Nearest Neighbors Regression, Multi-layer Perceptron regressor (using MLPRegressor from scikit-learn)
    - *Clustering*: K-Means clustering, Hierarchical clustering, (Agglomerative, Divisive), Density based methods (DBSCAN)
    - *Time Series forecasting*: ARMA, ARIMA, Machine Learning methods (e.g. Linear regression, support vector regression, Decision tree and random forest)
- **Evaluate the machine learning model**: Based on the category that the team's model falls into, they need to evaluate the model using appropriate evaluation metrics. Some suggestions for each category are provided as follows:
    - *Classification*: Accuracy, Precision, Recall, F1 Score, Confusion matrix, ROC (Receiver Operating Characteristic) curve, AUC-ROC (Area Under the ROC Curve)
    - *Regression*: Mean Absolute Error (MSE), Root Mean Squared Error (RMSE), R-squared
    - *Clustering*: Silhouette score, Inter-cluster distance, Using t-SNE (t-Distributed Stochastic Neighbor Embedding) and visualization
- **Enhance the model iteratively**: The team needs to improve the model by iteratively training, evaluating, and making adjustments. These adjustments could include the following steps:
    - Fine-tuning the hyperparameters (e.g., changing the number of trees in a random forest, changing the number of clusters in K-means clustering, changing the degrees of polynomials for polynomial regression, changing the depth of a random forest).

    - Improving the data processing step (adding new features, handling missing values differently, using another approach for data scaling, e.g., switching from MinMaxScaler to StandardScaler).
- **Select an appropriate machine learning algorithm**: The team needs to apply at least two different machine learning algorithms for their project. They improve each model by going through the cycle of training, improvement, and iterations. Then, considering both the model complexity and the performance obtained by each algorithm, the team chooses the best algorithm for their project (Note that one of these models could be a previously trained model by the team.)The following steps could be considered:
    - Choose the first algorithm to train a model. Train, evaluate, and iterate to improve the model performance in order to obtain the first optimal model.
    - Choose the second algorithm to train another model. Train, evaluate, and iterate to improve the model performance in order to obtain the second optimal model.
    - Compare the two models based on their performance and complexity. (For the performance evaluation at this stage, K-fold cross-validation is suggested.)
    - Choose the model that fits the project best.
- **Train a deep neural network**. The team needs to train a neural network for their problem. The neural network could be trained to address the following tasks:
    - Classification problem
    - Regression problem
    - Time series forecasting (using LSTM, GRU, RNN)
    - Training an autoencoder neural network for unsupervised learning or feature learning (i.e. to generate a list of features that can later be used for a recommender system or for clustering)

Assessment Guidelines

<u>Poor Criterion</u>

1- Clear individual contribution is documented.
For example:

Task / Student	Train	Evaluate	Iteration	Select Model	Neural Network
A	K-Means	Silhouette score, t-SNE visualization	Iterations to improve the K-means clustering	--------	--------
B	Agglomerative Clustering	Silhouette score, t-SNE visualization	Iterations to improve the Agglomerative clustering	--------	--------

C	------	-------	------	Evaluate and compare the two models.	Done

2- The team provides a rationale for why their problem is, for example, clustering rather than classification (a few sentences are sufficient)
3- The team has trained a classical model using scikit-learn.

Insufficient

1- The team chooses the appropriate metrics to evaluate the model (a brief explanation is required)
2- The team uses the test set to evaluate the model.
3- Training accuracy and test accuracy are compared.
4- Some discussions regarding overfitting or underfitting after the model evaluation are provided and seem rational.

Sufficient

1- The team performed iterations (training, evaluation, and adjustment) to improve the model's performance by experimenting with different hyperparameter values. The model's performance was reported using different figures or tables based on these hyperparameter values.
2- If the model has more than two hyperparameters, then at least two of these hyperparameters must go through the iterations.
3- In case the team considers data preprocessing to improve the model, those steps need to be explained. (Note that improving data preprocessing is not mandatory if the team has already enhanced the model through hyperparameter fine-tuning. However, if the team has not considered any hyperparameter fine-tuning, then this step becomes mandatory.)

Good

1- Another machine learning model is developed using a different machine learning algorithm.
2- For this model, iterations (training, evaluation, and adjustment) are applied to improve its performance.
3- Both models are evaluated based on an appropriate performance metric (using K-fold cross-validation is suggested for evaluating the performance of the two models).
4- Considering both model complexity and performance, the team chooses one model. Note that the chosen model is not always the one with the highest performance; model complexity also needs to be considered. Some explanation of how the team chose the preferred model needs to be provided.

Excellent

1- The team successfully defines and trains a neural network for their problem using available libraries (e.g., Keras, PyTorch).
2- Based on the problem (classification, regression, etc.), the structure of the neural network (e.g., the activation functions, number of neurons in the last layer) needs to be appropriate.
3- Learning curves for training and evaluation should be provided to demonstrate how well the model is learning from the training data and how well it generalizes to unseen data.
4- If they encounter overfitting or underfitting, the required steps to improve the model need to be implemented.
5- The final neural network model must be evaluated on the test dataset, using a suitable performance metric based on the problem.
6- A discussion comparing the neural network's performance to traditional machine learning models (obtained in the previous criteria) is required.