CODRELATE-2025

Whatsapp Link: https://chat.whatsapp.com/L9d7MJ6z5s6JMLqNJTENha

Round 2: Model Building

Problem Statement

Participants will use the preprocessed data from **Round 1** to build an efficient predictive model using **Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP), or Large Language Models (LLM)**. The goal is to develop a well-optimized solution that is **accurate, computationally efficient, interpretable, and user-friendly** with an innovative **frontend for visualization**.

Workflow / Methodology

1) Problem Definition & Model Selection

- Identify the type of problem (**Classification, Regression, Clustering, NLP, Time-Series Forecasting**).
- Choose the best-suited approach:
  - **ML Models** (Random Forest, XGBoost, SVM, etc.)
  - **DL Models** (CNNs for images, LSTMs for time series, Transformers for NLP)
  - **LLMs & NLP** (BERT, GPT-based models, Named Entity Recognition, Sentiment Analysis)

2) Data Preparation & Feature Engineering

- Handle feature selection and extraction to improve model performance.
- Use **dimensionality reduction** techniques (PCA, t-SNE) if necessary.
- Augment the dataset to **improve model generalization**.

3) Model Training & Optimization

- Implement **Hyperparameter Tuning** (GridSearchCV, RandomSearch, Bayesian Optimization).
- Apply **regularization techniques (L1, L2, Dropout)** to prevent overfitting.
- Use **cross-validation** to ensure robust model performance.

4) Computational Efficiency & Scalability

- Optimize memory usage and model efficiency using parallel computing.
- Implement **batch processing and GPU acceleration (TensorFlow/PyTorch)** for deep learning models.

5) Innovation & Creativity

- Explore innovative solutions using **AutoML, Transfer Learning, Custom Architectures**.
- Apply creative feature engineering techniques to boost accuracy.
- Integrate **multi-modal approaches** (e.g., combining text and image data).

6) Frontend Implementation & Model Integration

- Develop an **interactive frontend** to showcase model predictions.
- Use technologies like **Flask, FastAPI, Streamlit, React, or Dash** for deployment.

- Ensure an intuitive **UI/UX with graphs, dashboards, and dynamic visualizations**.

7) Model Evaluation & Explainability

- Select appropriate performance metrics:
  - **Classification:** Accuracy, Precision, Recall, F1-Score, AUC-ROC.
  - **Regression:** RMSE, MSE, R^2 Score.
  - **Clustering:** Silhouette Score, Davies-Bouldin Index.
- Implement **model explainability tools** (SHAP, LIME, Attention Mechanisms) to interpret results.

8) Final Report & Code Submission

- Submit a structured report with **problem definition, methodology, results, and key insights**.
- Provide clean, well-commented code with documentation.

Evaluation Criteria (100 Marks)

📌 Problem Understanding & Model Selection (15 points)

- Demonstrates clarity in defining the problem and choosing the most suitable **ML/DL/NLP/LLM** approach based on the task requirements.

📌 Feature Engineering & Data Preparation (15 points)

- Effectively handles **missing values**, performs **feature extraction**, and applies **dimensionality reduction** techniques to enhance model performance.

📌 Model Performance & Accuracy (20 points)

- Evaluates model success using key performance metrics such as **accuracy**, **RMSE**, **F1-score**, or other relevant metrics based on the problem type.

📌 Computational Efficiency & Optimization (10 points)

- Optimizes memory usage, performs **hyperparameter tuning**, and reduces model overhead to ensure efficient execution, especially with large datasets.

📌 Innovation & Creativity (10 points)

- Utilizes **novel approaches** in model building, such as **unique feature selection**, **creative transformations**, or **hybrid modeling techniques** that enhance model performance.

📌 Frontend Implementation & Visualization (15 points)

- Designs an intuitive and **well-structured frontend** that effectively showcases model predictions and presents insights through **visualizations**.

📌 Model Explainability & Interpretability (10 points)

- Provides justification for model predictions using **explainability tools** (e.g., SHAP, LIME), ensuring transparency and trustworthiness in the results.

📌 Report & Code Quality (10 points)

- Submits a **well-structured report** with clear documentation and **properly commented code** that explains the methodology, model choices, and results.