Systems Analysis & Design
Semester 2025-III
Course Project Description

Eng. Carlos Andrés Sierra, M.Sc.
Full-time Adjunct Professor
Computer Engineering Program
School of Engineering
Universidad Distrital Francisco José de Caldas

Overview

This course project integrates the primary lessons covered in previous workshops:

i. **Workshop #1: Analysis of a Kaggle Competition**.

ii. **Workshop #2: System Design and Architecture**.

iii. **Workshop #3: System Design Improvement and Management**.

iv. **Workshop #4: Computational Simulation and Validation**.

In this final phase, you will incorporate a **machine learning model** (such as a neural network or random forest) available on Kaggle to:

- Automatically generate or refine submissions for the competition you analyzed.
- Evaluate how these model-based solutions perform in real competition settings.
- Document your findings in a final report that references insights from prior workshops.

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Titular Professor at Universidad Distrital Francisco José de Caldas.

Any comment or concern about this document can be sent to Carlos A. Sierra at: *cavirguezs@udistrital.edu.co*.

Project Scope and Objectives

- **Integrate a Machine Learning Model:** Choose or fine-tune an available neural network or random forest (e.g., a specialized notebook or Kaggle kernel) that can assist with data preprocessing, feature engineering, or direct predictions.
- **End-to-End Pipeline:** Combine your analysis (Workshop #1), design decisions (Workshop #2), project management and improvement strategies (Workshop #3), and simulation approach (Workshop #4) into a coherent workflow.
- **Competition Submission:** Prepare at least one valid submission to the chosen Kaggle competition using your selected model.
- **Performance Evaluation:** Compare the model-driven submission to any baseline or traditional models used in previous workshops. Highlight improvements or drawbacks.
- **Project Report:** Summarize the entire process, referencing all prior workshop results and detailing the model application, Kaggle leaderboard insights, and recommended next steps.

Project Steps

1. **Model Setup:**
   - Identify a pre-trained neural network or random forest in Kaggle, or upload your own model version.
   - Justify the choice of model (e.g., domain relevance, performance on similar datasets, interpretability).
2. **Connector or API:**
   - Implement a straightforward *interface* or script to link your system design (Workshop #2) with the chosen model.
   - Ensure that data can be seamlessly fed into the model for inference, transformation, or other manipulations (e.g., generating predictions, feature engineering).
3. **Project Management and Improvement:**
   - Apply the management strategies and improvement actions defined in Workshop #3 to organize your workflow, track progress, and address challenges.
   - Document any changes made to the system design or model integration based on feedback, testing, or team decisions.
   - Use project management tools (e.g., Kanban boards, Gantt charts, issue trackers) to demonstrate how tasks were assigned, monitored, and completed.
   - Reflect on how iterative improvements impacted the overall system performance and project outcomes.

4. **Simulation and Testing:**

   - Reuse or extend the simulation strategies from Workshop #4 to evaluate how the model interacts with your system architecture.
   - Track performance (e.g., execution time, resource usage) and system behavior under varied scenarios (such as small data vs. large data).
   - Validate the robustness of your system by testing with edge cases, chaotic inputs, and unexpected user interactions.
   - Analyze the simulation results to assess the reliability, scalability, and efficiency of your integrated solution.

5. **Competition Submission:**

   - Generate predictions or post-processed solutions using the model, then submit results to the Kaggle competition.
   - Document the submission process, including data preparation, any hyperparameter tuning, and error handling.
   - Ensure that the submission adheres to Kaggle's guidelines and formats for successful evaluation.

6. **Performance and Discussion:**

   - Compare the model-based submission's score to previous approaches (traditional ML or deep learning methods).
   - Explain any disparities between expected performance and actual leaderboard results.
   - Propose potential next steps or model improvements.

7. **Final Report:**

   - Combine your writing from all workshops (analysis, design, management, simulation, model approach).
   - Include an **experiments section** detailing personal attempts to improve model outputs, data transformations, and relevant Kaggle codes.
   - Provide final remarks on the project's overall efficiency, user experience, and future research or development ideas.
   - Save your final PDF and relevant source codes in a dedicated folder (e.g., `Final_Course_Project`) in your GitHub repository.

Deliverables and Deadlines

- **Model Implementation Notebook or Script:** Show how the neural network or random forest was integrated (upload to Kaggle or store in your repository).
- **Final Report (PDF)**: Summarize the entire project, highlighting model design choices, Kaggle outcomes, and any code references.
- **Supporting Files:** Any Dockerfiles, `requirements.txt`, or environment.yml definitions for reproducibility.
- **Submission Deadline:** Refer to your course schedule or official announcements for the exact date and time.

Notes and Recommendations

- Coordinate with your team to ensure consistent integration of analysis, design, and simulation artifacts from prior workshops.
- Document any challenges faced when combining the model with your pipeline (e.g., large memory usage, slow inference times).
- If the competition results do not meet your expectations, discuss possible reasons and propose specific improvements (e.g., advanced feature engineering, more training epochs, or better structured data inputs).

This concludes your course project guidelines. By uniting analysis, design, simulation, and a neural network or random forest-based Kaggle submission, you will gain a rich, hands-on experience in modern systems engineering. Good luck!