AI TEAM TRAINING'26

TASK 3

Task 3.1: The F1 Spam Filter

The Scenario:

You are a data scientist for the Mercedes-AMG Petronas F1 Team. Your star driver, Hassan, is facing a critical issue: information overload. During intense races, his radio is flooded with a mix of vital commands from his race engineer, non-essential status updates, and erroneous "ghost" warnings from faulty sensors. This "radio spam" is shattering his focus, causing him to miss crucial apexes and lose valuable lap time. Your mission is to develop a filter that can intelligently distinguish between essential and non-essential messages.

Your Objective:

Your primary goal is to implement a Logistic Regression classifier from scratch using the Gradient Descent optimization algorithm. You will not be permitted to use any external machine learning libraries for the core implementation.

Upon completion, you will:

- Apply your custom-built Logistic Regression model to a real-world classification challenge on Kaggle.
- Assess your model's performance and generate submission files to be scored on the competition's leaderboard.

Technical Requirements:

To successfully complete this task, you must submit the following:

- Python files containing your from-scratch implementation of both the Gradient Descent algorithm and the Logistic Regression model.
- At least one successful submission to the designated Kaggle competition.

Bonus Objectives:

For those seeking an additional challenge and the opportunity to climb the leaderboard, consider the following enhancements:

- **Regularization:** Implement L1 (Lasso) and L2 (Ridge) regularization to improve your model's ability to generalize and prevent overfitting.
- **Advanced Gradient Descent:** Develop and compare the performance of different Gradient Descent variations, including:
    - Stochastic Gradient Descent (SGD)
    - Mini-batch Gradient Descent
- **Achieve a top 5 position on the leaderboard** using your manually implemented model.

Task 3.2: The F1 Spam Filter

The Scenario:

Formula 1 drivers push their bodies to the limit under extreme speeds, g-forces and race durations. Proper nutrition and precise calorie tracking are essential to keep them at peak performance.

In this competition, you are challenged to build a model that predicts the number of calories burned by F1 racers during a simulated training session. Using biometric and workout data derived from the Calories Burnt Prediction dataset, your predictions will help inform personalized diet plans for each driver.

Your Objective:

- Apply the required data preprocessing, according to your needs
- Create a regression model of your choice, you are free to experiment as much models as you want, but try at least 2 different models, where each model is in its own notebook/python file. Document all experiments, even bad performing ones. **Note you are allowed to use sci-kit learn or any other python library for the models, you are not supposed to implement them from scratch.**
- Train on train.csv (features + Calories target)
- Predict Calories on the unseen test.csv
- Optimize for lowest Root Mean Squared Error (RMSE)
- Submit your predictions to this contest
- Provide a pdf/markdown file that shows your experiments and their score. Experiments could be changing model, hyperparameters, features, etc.

Constraints:

- Data: biometric readings (heart rate, speed, duration, etc.) for each session
- Evaluation metric: RMSE, as shown in the Evaluation tab
- Whether you're a seasoned competitor or new to Kaggle, dive in.

Technical Requirements:

To successfully complete this task, you must submit the following:

- Python files/notebooks containing your models and preprocessing applied, multiple files in case of multiple models experiments.
- At least one successful submission to the designated Kaggle competition.
- PDF/Markdown Report documenting your experiments.

Bonus Objectives:

- **Achieve a top 5 position on the leaderboard** using your pipeline.
- Achieve higher results by **ensembling** your highest achieving models.