Introduction to Machine Learning
Challenge 1
Due Date: Nov 2, 2025 (11:55 PM)

Kaggle Competition Link:

https://www.kaggle.com/t/069d1cac492a4be1aa786f50d8c91122

Task 1: Fine-tuning of Decision Tree

1. Load the provided train1.csv from the competition.
2. Create your own train-validation split:
    - Use 70% of the data for training, 30% for validation.
    - Set random_state = YOUR_ERP_ID (e.g., random_state=123456).
    - Do not use the competition's hidden test set for tuning.
3. Experiment with a Decision Tree classifier and explore:
    - Feature selection: Use ANOVA (numeric) + Chi-Square (categorical) to filter features.
    - Wrapper-based selection: Apply forward selection and backward elimination to identify high-performing feature subsets. Given computational constraints, limit the search to a reasonable number of features.
    - Dimensionality reduction: Apply PCA and report how many components capture 90% variance?
    - Hyperparameter tuning: Use GridSearchCV or RandomizedSearchCV to tune max_depth, min_samples_split, class_weight, etc.
    - Feature importance: Extract and interpret top 10 important features.
4. Track your progress:
    - Baseline (default model) AUROC
    - After feature selection
    - After hyperparameter tuning
    - Final AUROC on your validation set
5. Deliverable: A short report (Report 1) showing your pipeline, key results, and insights.

Reminder: Your train-validation split (Step 2 above) for Task 1 must use your ERP ID as the random seed to ensure that everyone works on a slightly different dataset and discourages copying.

Task 2: Kaggle Submission

1. Load the full provided train.csv from the competition (do not use your Task 1 split).
2. Train your best-performing model on the entire training set (100% of train.csv).
3. You may use any model from standard ML libraries. But make sure that the models we have covered in the course must be evaluated, including:
   a. Categorical Naive Bayes
   b. K-Nearest Neighbors (KNN)
   c. Decision Tree
   d. Random Forest
   e. AdaBoost
4. Preprocess appropriately:
   a. Handle missing values
   b. Encode categorical variables (_cat columns)
   c. Apply scaling if needed (e.g., for KNN or Naive Bayes)
5. Generate predictions on the competition's test.csv and submit to Kaggle.
6. Deliverable: A detailed report that include:
   a. A comparison table of AUROC (on your own validation set) for all models tried
   b. Your final Kaggle public leaderboard score (AUROC)
   c. Analysis: Which model performed best? Why? (e.g., handling of imbalance, feature interactions, robustness)