AML-BASIC 2025 Project Implementation Guide

Yeast Protein Localization Case Study

Martina Castellucci

June 13, 2025

0. How to Use This Guide

This guide maps the complete implementation of the machine learning pipeline for the "Yeast Protein Localization" project to the core steps of the AML-BASIC 2025 course. It is intended to:

- Serve as a checklist for hands-on pipeline coverage
- Provide practical reference for core concepts and tools
- Demonstrate alignment between theoretical lectures and applied ML workflow

1. Project Overview

The project addresses the task of predicting the subcellular localization of proteins using the UCI Yeast dataset. All stages of the AML-BASIC pipeline are implemented, from data ingestion to model evaluation, with special attention to class imbalance and metric-driven interpretation.

2. Operational Mapping: Hands-on vs Project Steps

- **Hands-on 1 – Data Loading**
  Implemented with: `pandas.read_csv()`, `df.head()`, `df.dtypes`
  Objective: Dataset structure, feature types, initial label distribution
- **Hands-on 2–3 – EDA and Feature Exploration**
  Tools: Boxplots, barplots, Pearson correlation heatmap
  Purpose: Detect outliers, skewness, redundancy, class imbalance
- **Hands-on 3–4 – Data Preprocessing**
  Applied: `StandardScaler`, `LabelEncoder`, Stratified split (80/20)
  Rationale: Normalize feature space and preserve class proportions

- **Hands-on 5 – Imbalanced Classification**
  Applied: `SMOTE`, `class_weight="balanced"`
  Justification: Counteract class imbalance in training data

- **Hands-on 6–7 – Model Training and Tuning**
  Models: Logistic Regression, SVM, k-NN, Random Forest
  Tuning: `GridSearchCV` with 5-fold CV for Random Forest

- **Hands-on 7–9 – Evaluation and Visualization**
  Metrics: Accuracy, Macro F1, MCC, ROC-AUC, PR-AUC
  Visuals: Confusion matrices, One-vs-Rest ROC/PR curves, misclassification tables

3. Core Concepts Applied in Practice

Each theoretical concept introduced during lectures was implemented with matching tools and functions:

- Feature scaling and encoding with `StandardScaler` and `LabelEncoder`
- Handling class imbalance via SMOTE and class weighting
- Application of supervised models with diverse learning strategies
- Grid-based hyperparameter search using cross-validation
- Metric-based evaluation suitable for imbalanced multiclass problems
- Controlled experiment reproducibility using `random_state=42`

4. Summary of Coverage

- All pipeline stages fully implemented according to AML-BASIC
- Hands-on content directly mapped to project code and rationale
- Theoretical methods reflected in algorithm choices and metrics
- Results validated with plots, numerical scores, and interpretation

5. Final Remarks

This guide ensures that the project adheres to the intended design and structure of the AML-BASIC 2025 course.