# Machine Learning BASIC Report – Yeast Protein Localization Classification

Martina Castellucci
MSc in Bioinformatics, University of Bologna

Applied Machine Learning BASIC, 2024–2025

### Abstract

**Motivation:** Predicting the subcellular localization of proteins is essential for understanding their biological function and cellular dynamics. The Yeast dataset from the UCI repository presents a challenging classification task: 10 localization classes, severe class imbalance, and low-dimensional physicochemical features.

**Results:** A complete machine learning pipeline was implemented, including preprocessing, SMOTE oversampling, and classifier tuning via GridSearchCV. Random Forest achieved Accuracy = 0.62, Macro F1 = 0.58, and MCC = 0.51. ROC and PR curves confirmed performance on major classes; confusion matrices revealed key misclassifications. The notebook provides a robust ML template for biological classification under imbalance.

**Contact:** martina.castellucci@studio.unibo.it

**Supplementary materials:** `https://github.com/Martinaa1408/ML_basic_project/`

## Contents

## Glossary of Key Terms and Functions

- **SMOTE** – Synthetic Minority Over-sampling Technique: Method for generating synthetic samples of minority classes using nearest neighbors. Helps balance class distributions during training.
- **Stratified Split** – Train/test splitting strategy that maintains the proportion of each class. Prevents underrepresentation of rare classes in test set.
- **StandardScaler** – A preprocessing tool from `sklearn.preprocessing` that scales each feature to zero mean and unit variance, improving convergence and model performance.
- **RandomForestClassifier** – Ensemble method from `sklearn.ensemble` that builds multiple decision trees and aggregates their outputs. Handles non-linearities and works well with imbalanced data.
- **GridSearchCV** – Model selection utility from `sklearn.model_selection` that performs exhaustive search over hyperparameter values using cross-validation.
- **class_weight="balanced"** – A parameter in many scikit-learn classifiers that automatically adjusts class weights inversely proportional to class frequencies.
- **Macro F1-score** – Average of per-class F1-scores, treating each class equally. Useful for imbalanced multiclass problems.
- **MCC (Matthews Correlation Coefficient)** – Balanced evaluation metric that accounts for all confusion matrix categories. Robust under severe imbalance.
- **PR Curve** – Precision–Recall curve: plots precision vs. recall at different thresholds. Better than ROC when evaluating imbalanced data.
- **ROC-AUC** – Area under the Receiver Operating Characteristic curve: measures classifier's ability to distinguish between classes in a One-vs-Rest setting.
- **Confusion Matrix** – Tabular summary showing predicted vs. actual classes. Useful for identifying specific misclassifications.
- **OneVsRestClassifier** – Strategy that builds one classifier per class, treating all other classes as negative. Used for multiclass classification with binary base estimators.

# 1 Introduction

## 1.1 Biological background

Subcellular localization is a fundamental property of proteins that determines where in the cell a protein performs its function. Mislocalization can impair function and lead to pathological states. For these reasons, predicting localization has become a crucial task in computational biology.

Several datasets exist to support this task. Among them, the Yeast dataset from the UCI repository is widely used as a benchmark for multiclass protein localization prediction. It provides a representative example of the challenges typically encountered in biological data: low feature dimensionality, noisy signals, and strong class imbalance.

## 1.2 Machine learning challenges in multiclass protein localization

From a machine learning perspective, this dataset introduces a number of challenges:

- **Multiclass classification**: Unlike binary classification, this problem involves 10 output classes with varying representation.

- **Severe class imbalance**: Most algorithms will overfit on the dominant classes and ignore rare ones.
- **Feature sparsity**: Only 8 numeric features, all continuous and derived from sequence analysis tools.
- **Biological noise**: Some features are predictive only in specific contexts (e.g., N-terminal signal prediction), which may not generalize across all localizations.

Despite these limitations, the dataset remains an excellent benchmark for comparing ML techniques in imbalanced multiclass settings. In this work, we aim to build a robust pipeline that addresses these issues using class-weighting, SMOTE oversampling, and fair metric-based evaluation.

## 2 Aim of the Study

- Develop a pipeline for multiclass protein localization
- Address class imbalance using SMOTE and class weights
- Compare Logistic Regression, SVM, k-NN, Random Forest
- Evaluate with macro F1, MCC, PR/ROC-AUC, confusion matrices

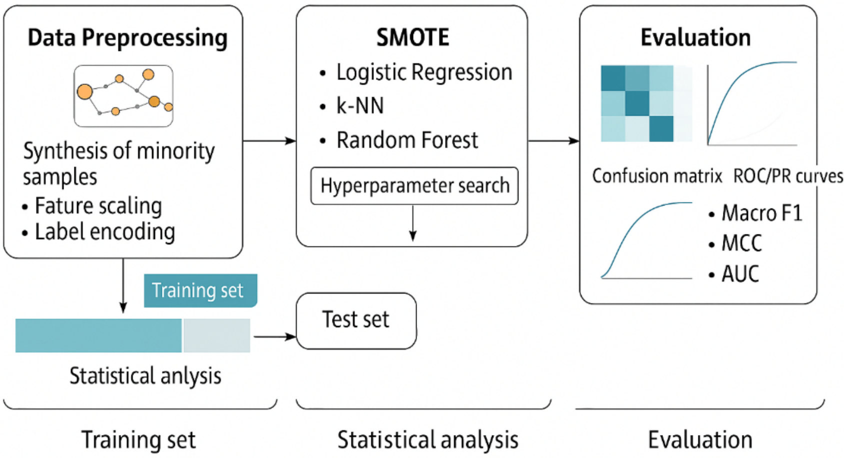


Figure 1: Schematic representation of the ML pipeline for protein localization, including preprocessing, balancing, classification, and evaluation.

## 3 Materials and Methods

### 3.1 Dataset Description

The dataset used in this project is the Yeast dataset from the UCI Machine Learning Repository. It includes 1,484 proteins from *Saccharomyces cerevisiae*, each described by 8 numerical features derived from amino acid sequences. These features reflect biologically relevant properties related to protein targeting and sorting, such as:

- `mcg`, `gvh`, `alm`: signal peptide predictors;
- `mit`: mitochondrial targeting signal score;

- `erl`: binary feature for the ER retention motif (HDEL);
- `pox`, `vac`, `nuc`: discriminant scores for peroxisomal, vacuolar, and nuclear localization.

Each protein is labeled with one of 10 subcellular localization classes: `CYT`, `NUC`, `MIT`, `ME1`, `ME2`, `ME3`, `POX`, `ERL`, `EXC`, and `VAC`. The class distribution is highly skewed, with CYT and NUC comprising most examples, while ERL and POX contain fewer than 5 instances.

### 3.2 Preprocessing

The preprocessing pipeline aimed to prepare the data for robust training and evaluation. All feature columns were scaled using `StandardScaler` to normalize the input range and prevent dominance of high-magnitude variables. The class labels, originally encoded as strings, were converted into integer format using `LabelEncoder` from scikit-learn.

To preserve the distribution of the original classes, especially in the presence of class imbalance, we employed a stratified 80/20 split for training and testing. This ensured that all 10 localization classes were proportionally represented in both sets. Additionally, we addressed class imbalance using two complementary strategies. First, we applied the `class_weight="balanced"` parameter to classifiers that supported it. Second, we applied SMOTE (Synthetic Minority Oversampling Technique) to the training set to synthetically generate samples for minority classes. The number of neighbors used by SMOTE was dynamically adjusted depending on the size of the smallest class, and classes with extremely low support were excluded from oversampling to avoid overfitting on synthetic noise. To ensure full reproducibility, all randomized operations — including train/test splitting, SMOTE resampling, and model initialization — were performed using a fixed random seed (`random_state=42`).

### 3.3 Model training and tuning

Four supervised learning algorithms were employed to tackle the classification task: Logistic Regression, k-Nearest Neighbors (k-NN), Support Vector Machine (SVM) with a linear kernel, and Random Forest. These models were selected to represent both linear and non-linear decision strategies, as well as distance-based learning.

Random Forest, which yielded the best preliminary results, was further optimized using `GridSearchCV`. The grid search explored different values for key hyperparameters, including `n_estimators`, `max_depth`, and `min_samples_split`. Cross-validation with 5 folds ensured a reliable and generalized estimate of performance during tuning.

### 3.4 Evaluation metrics

The following metrics were used to evaluate the models:

**Accuracy**: Overall correctness.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

**Precision**: True positive rate among predicted positives.

$$Precision = \frac{TP}{TP + FP}$$

**Recall**: Fraction of true positives retrieved.

$$Recall = \frac{TP}{TP + FN}$$

**F1-score**: Harmonic mean of precision and recall.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

**Macro F1**: Unweighted mean of F1 scores across all classes, treating each class equally.

**Weighted F1**: Mean of F1 scores weighted by the number of true instances per class.

**MCC (Matthews Correlation Coefficient)**: A robust metric even under severe class imbalance.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

**ROC-AUC**: Measures class separability by calculating the area under the Receiver Operating Characteristic curve in a one-vs-rest setting.

**PR-AUC**: Captures the precision-recall trade-off, especially informative under high class imbalance.

# 4 Results

This section presents the outcomes of the machine learning pipeline applied to the yeast protein localization dataset. The classification performance was evaluated on the test set using several metrics and visual diagnostics, including the confusion matrix and ROC/PR curves. The train/test split used was stratified (80/20), ensuring proportional representation of all 10 classes in both sets. Model selection was performed using only the training set. The results presented refer to a single evaluation on the test set and are not averaged across multiple runs.

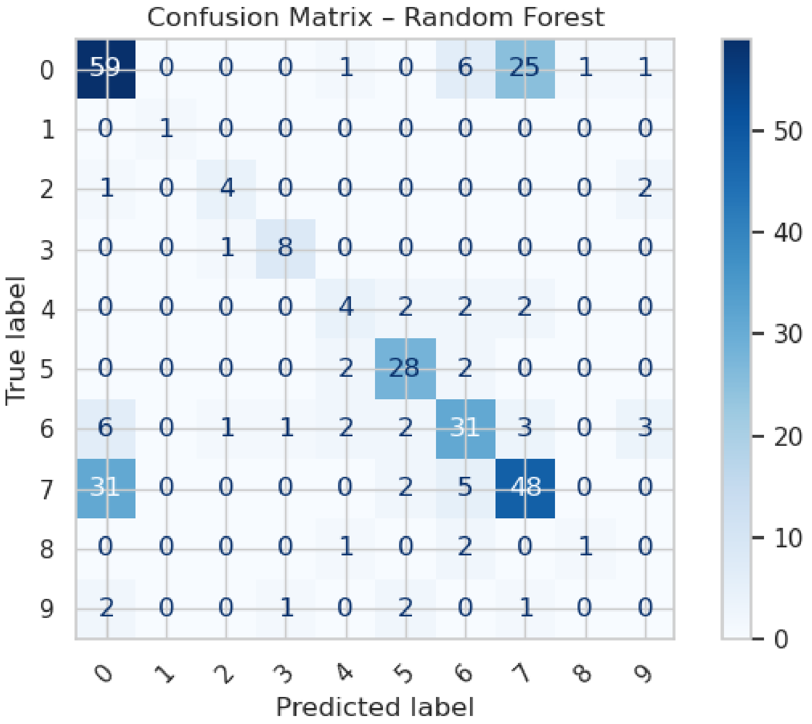
## 4.1 Model performance comparison

| Model | Accuracy | Macro F1 | MCC |
|---|---|---|---|
| Logistic Regression | 0.51 | 0.49 | 0.40 |
| Random Forest | 0.62 | 0.58 | 0.51 |
| SVM | 0.55 | 0.57 | 0.44 |
| k-NN | 0.47 | 0.44 | 0.35 |

Table 1: Performance metrics on the test set.

All reported metrics refer to the held-out test set after model selection via 5-fold stratified cross-validation. This ensures that hyperparameter tuning did not bias the evaluation and that performance estimates are robust and generalizable.

## 4.2 Confusion matrix



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 59 | 0 | 0 | 0 | 1 | 0 | 6 | 25 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 3 | 0 | 0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 4 | 2 | 2 | 2 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 2 | 28 | 2 | 0 | 0 | 0 |
| 6 | 6 | 0 | 1 | 1 | 2 | 2 | 31 | 3 | 0 | 3 |
| 7 | 31 | 0 | 0 | 0 | 0 | 2 | 5 | 48 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 |
| 9 | 2 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 |

Figure 2: Confusion matrix for Random Forest (tuned)

## 4.3 ROC and PR curves

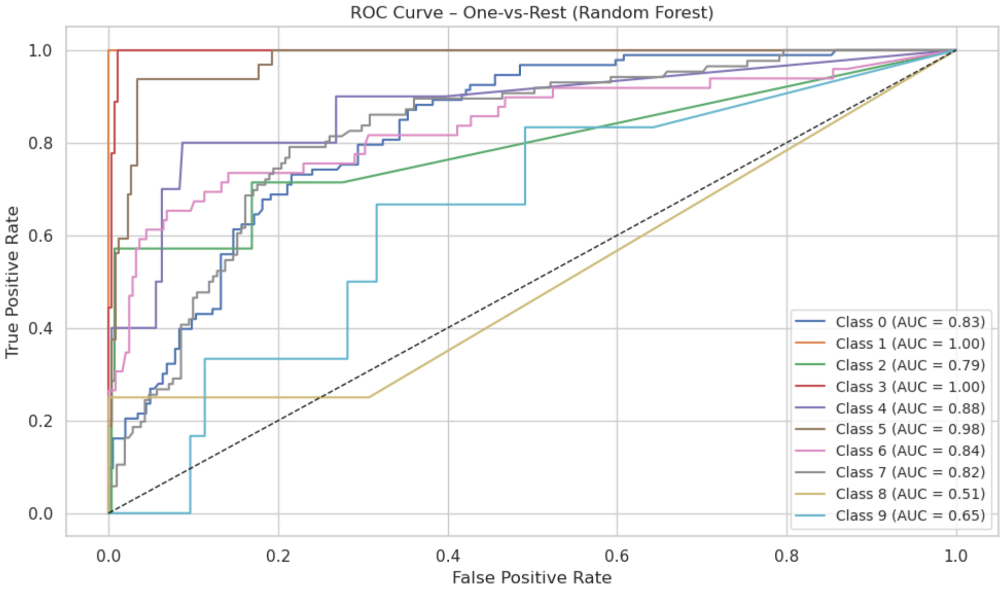


Figure 3: ROC Curves – One-vs-Rest for all classes.

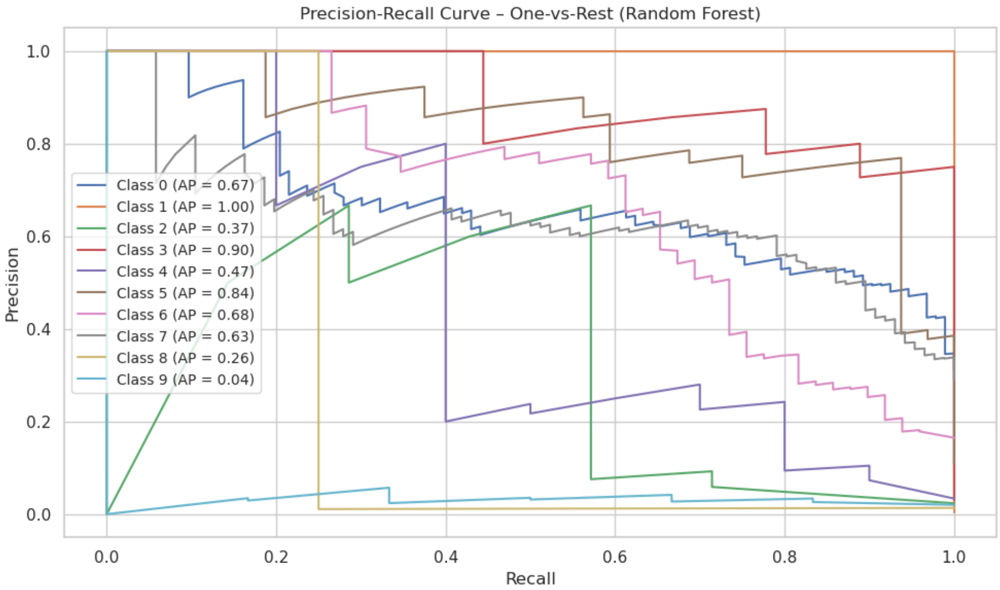


Figure 4: Precision-Recall Curves – One-vs-Rest for all classes.

## 5 Discussion

The results obtained in this study highlight the multifaceted impact of class imbalance on multiclass classification in bioinformatics. Among the models tested, the Random Forest classifier, when optimized via grid search, consistently outperformed others in terms of both accuracy and fairness across classes. It achieved a macro F1-score of 0.58 and an overall accuracy of 0.62, demonstrating its robustness in handling the non-linear and noisy nature of the Yeast dataset.

From the confusion matrix analysis, it became evident that classes with a higher representation in the training set, such as `CYT`, `NUC`, and `MIT`, were classified with high precision and recall. In contrast, rare classes such as `ERL` and `POX` suffered from extremely low recall, often being misclassified into the more frequent categories. While the application of SMOTE led to an improved recall for moderately underrepresented classes like `EXC` and `VAC`, its effectiveness was limited for ultra-rare classes. This is likely due to the inability of synthetic oversampling to compensate for the lack of original signal variation in extremely small classes.

ROC and PR curve analyses further clarified these trends. ROC-AUC values for frequent classes ranged from 0.85 to 0.90, confirming good separability. However, rare classes like `ERL` reached an AUC of 1.00—an artefact of having only one positive sample in the test set. This artificially high score illustrates how ROC metrics can be misleading in imbalanced scenarios. Conversely, PR curves provided a more realistic view: while common classes exhibited high average precision, rare classes had AP scores close to zero, clearly exposing model limitations.

Importantly, the macro F1 and MCC metrics served as reliable indicators of overall performance, being less biased toward dominant classes compared to accuracy. The evaluation confirmed that while accuracy may appear acceptable, it masks disparities in per-class behavior that are critical in biological interpretation.

Biologically, misclassifications between `CYT` and `MIT` are plausible, given their overlapping signal features. This reinforces the need for domain knowledge in interpreting machine learning outputs and for potentially enriching the feature space with structural or contextual annotations.

**Future work** should focus on extending the pipeline by:

- Incorporating protein embeddings from language models (e.g., ProtT5, ESM)
- Exploring ensemble models beyond Random Forest (e.g., LightGBM, CatBoost)

- Integrating protein domain annotations or sub-sequence motifs
- Applying calibration and uncertainty quantification to better handle low-confidence predictions

In conclusion, this project demonstrated how classical ML techniques, when combined with thoughtful preprocessing and evaluation, can produce interpretable and robust results on challenging biological classification problems. The pipeline developed here offers a scalable and reproducible approach to similar tasks in computational biology.

*For detailed visualizations, including feature distributions, ROC and PR curves, and class-specific error analyses, please refer to the Supplementary Material.*

## Acknowledgments

- Dataset: UCI ML Repository `https://archive.ics.uci.edu/dataset/110/yeast`
- Course: AML-BASIC 2025 (Prof. Bonacorsi, Clissa, UniBO)
- course notebook `https://drive.google.com/drive/folders/1ZrQpF_F9E45yQTO9mG8Izr3LaECVH0aH`

## References

- Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research, 16, 321–357.
  `https://doi.org/10.1613/jair.953`
- Horton, P., Nakai, K. (1997). *Better prediction of protein cellular localization sites with the k nearest neighbors classifier.* Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (ISMB), 5, 147–152.
  `https://pubmed.ncbi.nlm.nih.gov/9322029/`
- scikit-learn documentation: `https://scikit-learn.org`
- imbalanced-learn documentation: `https://imbalanced-learn.org`