PYTHON CHO KHOA HỌC DỮ LIỆU

Giảng viên: Hà Minh Tuấn

THAM SỐ MẶC ĐỊNH CỦA ModelTrainer

Nhóm thực hiện:

Đặng Đỉnh Đoàn 23280046
Phạm Thanh Uy 23280097

TP. Hồ Chí Minh, tháng 12 năm 2025

Mục lục

1 Giới thiệu

Dây là các tham số mặc định của class ModelTrainer, khi User không truyền vào, sẽ tự động lấy các tham số này.

2 Nhóm cấu hình chính – TrainerConfig

```
@dataclass
class TrainerConfig:
    target_col: str = "Outcome"
    random_state: int = 42
    scoring_primary: str = "f1"
    scoring_other: List[str] = ["roc_auc", "accuracy", "precision", "recall"]
    cv_splits: int = 5
    use_randomized_search: bool = True
    n_iter_random_search: int = 20
    use_smote: bool = True
    model_names: List[str] = ["log_reg", "random_forest"]
    model_output_dir: Optional[Union[str, Path]] = None
```

2.1 Target và random state

- **target_col**
  - Hiện tại: "Outcome"
  - Có thể đổi theo dataset khác: "Label", "HasDiabetes", "Target", ...
- **random_state**
  - Hiện tại: 42
  - Có thể đổi sang bất kỳ số nguyên: 0, 123, 2024, ...
  - Nên cố định để kết quả reproducible

2.2 Mục tiêu tối ưu và metrics

- **scoring_primary**
  - Hiện tại: "f1"
  - Các option điển hình của scikit-learn:
    - "roc_auc", "accuracy", "precision", "recall", "balanced_accuracy"
    - Hoặc custom scorer nếu tự định nghĩa
- **scoring_other**
  - Hiện tại: ["roc_auc", "accuracy", "precision", "recall"]
  - Có thể thêm/bớt metric phụ:
    - Bớt: ["roc_auc", "accuracy"]

        * Thêm: ["roc_auc", "accuracy", "precision", "recall", "balanced_accuracy"]
    - Lưu ý: phải là tên metric mà sklearn hỗ trợ

2.3 Cross-validation & search

- **cv_splits**
    - Hiện tại: 5
    - Có thể đổi: 3, 4 (nhanh hơn), 10 (ổn định hơn nhưng chậm)
- **use_randomized_search**
    - True: dùng RandomizedSearchCV
    - False: dùng GridSearchCV
- **n_iter_random_search**
    - Hiện tại: 20
    - Chỉ áp dụng khi use_randomized_search=True
    - Có thể giảm: 10 (nhanh hơn), tăng: 50 (tìm kỹ hơn, chậm hơn)

2.4 Imbalance & danh sách model

- **use_smote**
    - Hiện tại: True
    - True: dùng SMOTE để cân bằng nhãn
    - False: không dùng SMOTE, rely vào class_weight='balanced'
- **model_names**
    - Hiện tại: ["log_reg", "random_forest"]
    - Có thể chỉnh:
        * Chỉ Logistic: ["log_reg"]
        * Chỉ Random Forest: ["random_forest"]
        * Mở rộng: ["log_reg", "random_forest", "xgboost"], ["log_reg", "svm"], ...

2.5 Thư mục lưu model

- **model_output_dir**
    - Hiện tại: None → không lưu model
    - Có thể đặt:
        * "models"→ <project_root>/models
        * "artifacts/models"→ <project_root>/artifacts/models

    * Đường dẫn tuyệt đối: r"E:⊨ ”

3 Nhóm “núm vặn” ẩn trong method

3.1 run_training(...)

- X_test, y_test
  - Nếu truyền: tính metrics test, experiments.csv, best_model.txt
  - Nếu None: chỉ CV trên train, không có test metrics
- n_jobs
  - Hiện tại: n_jobs=1
  - Có thể đổi: -1 để dùng hết core
- verbose
  - Hiện tại: 1
  - Có thể tăng: 2 để log chi tiết

3.2 _get_model_and_param_grid(...)

- **Logistic Regression**
  - solver: "liblinear"(hiện dùng), "lbfgs", "saga"
  - max_iter: 1000 (có thể tăng 2000)
  - class_weight: "balanced"/ None
  - param_grid:
    * C: [0.01, 0.1, 1.0, 10.0] (có thể thêm log-scale)
    * penalty: ["l1", "l2"] (phụ thuộc solver)
- **Random Forest**
  - n_estimators: 200 (tăng 300, 500 nếu muốn)
  - class_weight: "balanced"/ None
  - n_jobs: 1 (có thể -1)
  - param_grid:
    * n_estimators: [100, 200, 300]
    * max_depth: [None, 5, 10, 20]
    * min_samples_split: [2, 5]
- **SMOTE**
  - True: Pipeline(SMOTE + model), prefix "model__"cho param_grid

  - False: model gốc, param_grid không có prefix

3.3 evaluate_on_test(...)

- Metrics hiện tại: accuracy, precision, recall, f1, roc_auc, tn, fp, fn, tp
- Có thể thêm: specificity = tn / (tn+fp), balanced_accuracy, ...
- zero_division: 0 (có thể đổi 1 hoặc "warn")

3.4 compute_shap_values(...)

- max_samples
  - Default: 200
  - Khi gọi: trainer.compute_shap_values("log_reg", X_train, max_samples=500)
  - Ít → nhanh hơn, nhiều → ổn định hơn nhưng chậm
- Lớp multi-class
  - Nếu sv.ndim=3 và sv.shape[1]=2 → lấy lớp positive (index 1)
  - Ngược lại → average theo axis=1

3.5 Lưu / load model

- _maybe_save_model(...)
  - Tên file: model_output_dir / f"{model_name}_best.joblib"
  - Có thể đổi pattern: f"{model_name}_best_{scoring_primary}.joblib"
- load_saved_model(model_name, path=None)
  - Truyền path cụ thể nếu muốn load khác
  - path=None → dùng <model_output_dir>/<model_name>_best.joblib

3.6 RESULTS_DIR

- Thư mục results dùng trong:
  - experiments.csv, best_model.txt
- Có thể chỉnh trong `src/utils/config.py`:
  - Đổi RESULTS_DIR sang Path("outputs"), Path("reports"), ...