Отчет по Заданию №3: Классификация данных ЭКГ с использованием AutoML

关于任务 3 的报告：使用 AutoML 进行心电图数据分类

Айдана Халихаз
阿依达娜 · 哈力哈孜

30 апреля 2025 г.
2025 年 4 月 30 日

1 Цель работы

Целью данного задания является обработка кардиологического датасета для решения задачи бинарной классификации статуса здоровья (’Healthy_Status’) на основе параметров электрокардиограммы (ЭКГ).Датасет был загружен из репозитория GitHub: https://github.com/AI-is-out-there/data2lab.git. Для анализа было взято первые 5000 строк данных.

Была сформирована выборка данных, включающая следующие столбцы: `'Count_subj'`, `'rr_interval'`, `'p_end'`, `'qrs_onset'`, `'qrs_end'`, `'p_axis'`, `'qrs_axis'`,

1 目标

本任务的目标是处理一个心脏病学数据集，以解决基于心电图（ECG）参数对健康状况（’Healthy_Status’）进行二元分类的问题。数据集来源于 GitHub 仓库：https://github.com/AI-is-out-there/data2lab.git。分析选取了数据集的前 5000 行。

根据要求，构建的训练样本包含以下列：[’Count_subj’, ’rr_interval’, ’p_end’, ’qrs_onset’, ’qrs_end’, ’p_axis’, ’qrs_axis’, ’t_axis’, ’Healthy_Status’]。

关键任务是研究并应用 AutoML（自动化机器学习）框架来构建分类器，使用 F1 分数（F1-metric）评估其

't_axis', 'Healthy_Status'.

Ключевой задачей является исследование и применение фреймворков AutoML для построения классификатора, оценка его точности с использованием F1-метрики и построение матрицы ошибок (confusion matrix).

2 Реализация с использованием AutoML (PyCaret)

2.1 Загрузка и подготовка данных

Данные были загружены с использованием библиотеки pandas из указанного URL. Были отобраны только необходимые столбцы и первые 5000 записей согласно условию задачи.

2.2 Выбор и применение фреймворка AutoML

Для решения задачи был выбран фреймворк AutoML PyCaret. PyCaret является библиотекой машинного обучения с низким уровнем кода (low-code), которая автоматизирует рабочие процессы МО. Она позволяет быстро сравнивать

准确性，并构建混淆矩阵（confusion matrix）。

2 使用 AutoML（PyCaret）的算法实现

2.1 数据加载与准备

使用 pandas 库从指定的 URL 加载了数据。根据任务要求，仅选择了必需的列和前 5000 条记录。

2.2 AutoML 框架的选择与应用

为了完成此任务，我们选择了 PyCaret AutoML 框架。PyCaret 是一个低代码（low-code）机器学习库，可以自动化机器学习工作流程。它能够快速比较多种模型、执行数据预处理和超参数调整，非常适合本任务中要求的探索和快速原型设计。

различные модели, выполнять предобработку данных и настройку гиперпараметров. Это делает ее подходящим выбором для исследования и быстрого прототипирования согласно заданию.

2.3 Настройка среды PyCaret и сравнение моделей

Среда PyCaret была инициализирована с помощью функции setup. В качестве целевой переменной (target) был указан столбец 'Healthy_Status'. Были применены нормализация данных (normalize=True) и метод для борьбы с дисбалансом классов (`fix_imbalance=True`). PyCaret автоматически разделяет данные на обучающую и тестовую выборки (по умолчанию 70/30).

Затем была использована функция compare_models для обучения и сравнения различных моделей классификации с использованием кросс-валидации. Модели были отсортированы по F1-метрике (sort='F1'), чтобы выбрать лучшую модель на основе этого показателя.

2.3 PyCaret 环境设置与模型比较

使用 setup 函数初始化了 PyCaret 环境。目标变量（target）指定为 `Healthy_Status` 列。应用了数据归一化（`normalize=True`）和处理类别不平衡的方法（`fix_imbalance=True`）。PyCaret 自动将数据划分为训练集和测试集（默认为 70/30）。

接着，使用 compare_models 函数通过交叉验证来训练和比较多种分类模型。这些模型按照 F1 分数（sort='F1'）进行排序，以便根据该指标选出最佳模型。

2.4 最佳模型评估

上一步选出的最佳模型，使用 predict_model 函数在预留的测试集（hold-out set）上进行了评估。该函数会自动计算包括 F1 分数在内的多种性能指标，并构建混淆矩阵。

2.4 Оценка лучшей модели

Лучшая модель, выбранная на предыдущем шаге, была оценена на отложенной тестовой выборке (hold-out set) с помощью функции predict_model. Эта функция автоматически рассчитывает различные метрики производительности, включая F1-метрику, и строит матрицу ошибок.

3 Результаты

При выполнении кода compare_models PyCaret представляет таблицу с результатами кросс-валидации для всех протестированных моделей, отсортированную по F1-метрике. Это позволяет определить, какая модель показала наилучшие результаты на обучающих данных. Имя лучшей модели выводится в консоль.

...

3 结果

执行 compare_models 代码时，PyCaret 会展示一个表格，其中包含所有测试模型的交叉验证结果，并按 F1 分数排序。这有助于确定哪个模型在训练数据上表现最佳。最佳模型的名称会输出到控制台。

...

	Description	Value
0	Session id	123
1	Target	Healthy_Status
2	Target type	Binary
3	Original data shape	(5000, 9)
4	Transformed data shape	(7156, 9)
5	Transformed train set shape	(5656, 9)
6	Transformed test set shape	(1500, 9)
7	Numeric features	8
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fix imbalance	True
13	Fix imbalance method	SMOTE
14	Normalize	True
15	Normalize method	zscore
16	Fold Generator	StratifiedKFold
17	Fold Number	10
18	CPU Jobs	-1
19	Use GPU	False
20	Log Experiment	False
21	Experiment Name	clf-default-name
22	USI	dbb4

Figure 1: Заполнитель для таблицы сравнения моделей PyCaret

	Description	Value
0	Session id	123
1	Target	Healthy_Status
2	Target type	Binary
3	Original data shape	(5000, 9)
4	Transformed data shape	(7156, 9)
5	Transformed train set shape	(5656, 9)
6	Transformed test set shape	(1500, 9)
7	Numeric features	8
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fix imbalance	True
13	Fix imbalance method	SMOTE
14	Normalize	True
15	Normalize method	zscore
16	Fold Generator	StratifiedKFold
17	Fold Number	10
18	CPU Jobs	-1
19	Use GPU	False
20	Log Experiment	False
21	Experiment Name	clf-default-name
22	USI	dbb4

Figure 1: PyCaret 模型比较表格

Функция predict_model, примененная к лучшей модели, генерирует отчет об оценке на тестовой выборке. Этот отчет включает:

1. Матрицу ошибок (Confusion Matrix): Визуальное представление точности классификатора, показывающее количество истинно положительных, истинно отрицательных, ложно положительных и ложно отрицательных предсказаний.

2. Таблицу метрик производительности:

应用于最佳模型的 predict_model 函数会生成在测试集上的评估报告。该报告包含：

1. 混淆矩阵 (Confusion Matrix): 分类器准确性的可视化表示，显示真正例、真负例、假正例和假负例的数量。

2. 性能指标表: 包括测试集上的准确率（Accuracy）、AUC、召回率（Recall）、精确率（Precision）、F1 分数（F1-score）、Kappa 和 MCC。

Включая Accuracy, AUC, Recall, Precision, F1-метрику, Kappa, MCC для тестового набора данных.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.8549	0.9189	0.8067	0.5897	0.6808	0.5899	0.6024	6.5710
gbc	Gradient Boosting Classifier	0.8400	0.9139	0.8558	0.5546	0.6726	0.5731	0.5971	0.9840
et	Extra Trees Classifier	0.8483	0.9141	0.7858	0.5770	0.6651	0.5700	0.5816	0.5590
ada	Ada Boost Classifier	0.8320	0.9136	0.8677	0.5399	0.6650	0.5611	0.5897	0.4210
rf	Random Forest Classifier	0.8517	0.9174	0.7680	0.5873	0.6649	0.5719	0.5810	1.2470
lightgbm	Light Gradient Boosting Machine	0.8486	0.9151	0.7412	0.5838	0.6525	0.5576	0.5646	0.6940
xgboost	Extreme Gradient Boosting	0.8460	0.9098	0.7144	0.5804	0.6402	0.5436	0.5486	0.1810
dt	Decision Tree Classifier	0.8343	0.7681	0.6607	0.5598	0.6048	0.5011	0.5048	0.0970
svm	SVM - Linear Kernel	0.6234	0.8073	0.9717	0.3375	0.4996	0.2997	0.4092	0.0590
lr	Logistic Regression	0.6094	0.8026	0.9926	0.3298	0.4948	0.2900	0.4083	0.8810
knn	K Neighbors Classifier	0.6583	0.7874	0.8440	0.3429	0.4872	0.2945	0.3621	0.0950
qda	Quadratic Discriminant Analysis	0.5854	0.8965	0.9985	0.3172	0.4812	0.2677	0.3921	0.0480
nb	Naive Bayes	0.5389	0.8904	0.9985	0.2945	0.4546	0.2245	0.3545	0.0480
lda	Linear Discriminant Analysis	0.5091	0.7207	1.0000	0.2835	0.4411	0.2018	0.3334	0.0730
ridge	Ridge Classifier	0.5023	0.7215	1.0000	0.2793	0.4364	0.1944	0.3274	0.0480
dummy	Dummy Classifier	0.8080	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0500

Figure 2: Матрицы Ошибок (Confusion Matrix)

在保留测试集上评估最佳模型：

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	CatBoost Classifier	0.8480	0.9135	0.7882	0.5761	0.6657	0.5704	0.5821

Figure 3: таблицы метрик из predict_model

Ключевые результаты, требуемые заданием - F1-метрика и матрица ошибок - автоматически генерируются и отображаются функцией predict_model при выполнении кода. F1-метрика позволяет оценить баланс между точностью (precision) и полнотой (recall) классификатора, что особенно важно при работе с несбалансированными данными, как это часто бывает в медицинских задачах. Матрица ошибок дает детальное представление о типах ошибок, допускаемых моделью.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.8549	0.9189	0.8067	0.5897	0.6808	0.5899	0.6024	6.5710
gbc	Gradient Boosting Classifier	0.8400	0.9139	0.8558	0.5546	0.6726	0.5731	0.5971	0.9840
et	Extra Trees Classifier	0.8483	0.9141	0.7858	0.5770	0.6651	0.5700	0.5816	0.5590
ada	Ada Boost Classifier	0.8320	0.9136	0.8677	0.5399	0.6650	0.5611	0.5897	0.4210
rf	Random Forest Classifier	0.8517	0.9174	0.7680	0.5873	0.6649	0.5719	0.5810	1.2470
lightgbm	Light Gradient Boosting Machine	0.8486	0.9151	0.7412	0.5838	0.6525	0.5576	0.5646	0.6940
xgboost	Extreme Gradient Boosting	0.8460	0.9098	0.7144	0.5804	0.6402	0.5436	0.5486	0.1810
dt	Decision Tree Classifier	0.8343	0.7681	0.6607	0.5598	0.6048	0.5011	0.5048	0.0970
svm	SVM - Linear Kernel	0.6234	0.8073	0.9717	0.3375	0.4996	0.2997	0.4092	0.0590
lr	Logistic Regression	0.6094	0.8026	0.9926	0.3298	0.4948	0.2900	0.4083	0.8810
knn	K Neighbors Classifier	0.6583	0.7874	0.8440	0.3429	0.4872	0.2945	0.3621	0.0950
qda	Quadratic Discriminant Analysis	0.5854	0.8965	0.9985	0.3172	0.4812	0.2677	0.3921	0.0480
nb	Naive Bayes	0.5389	0.8904	0.9985	0.2945	0.4546	0.2245	0.3545	0.0480
lda	Linear Discriminant Analysis	0.5091	0.7207	1.0000	0.2835	0.4411	0.2018	0.3334	0.0730
ridge	Ridge Classifier	0.5023	0.7215	1.0000	0.2793	0.4364	0.1944	0.3274	0.0480
dummy	Dummy Classifier	0.8080	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0500

Figure 2: 混淆矩阵（Confusion Matrix）

在保留测试集上评估最佳模型：

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	CatBoost Classifier	0.8480	0.9135	0.7882	0.5761	0.6657	0.5704	0.5821

Figure 3: 在保留测试集上评估最佳模型

任务要求的关键结果——F1 分数和混淆矩阵——在执行代码时由 predict_model 函数自动生成和显示。F1 分数评估了分类器的精确率（precision）和召回率（recall）之间的平衡，这在处理不平衡数据集（如医疗任务中常见的那样）时尤为重要。混淆矩阵则详细展示了模型所犯错误的类型。

4 Заключение

Задача по обработке кардиологического датасета и построению бинарного классификатора с использованием AutoML была успешно выполнена. Был использован фреймворк PyCaret для автоматизации процесса выбора и оценки моделей. Определена лучшая модель на основе F1-метрики, и для нее были рассчитаны F1-показатель и построена матрица ошибок на тестовой выборке, что соответствует требованиям задания.

Использование PyCaret позволило эффективно сравнить множество моделей и получить оценку производительности классификатора с минимальными усилиями по написанию кода.

4 结论

使用 AutoML 处理心脏病学数据集并构建二元分类器的任务已成功完成。利用 PyCaret 框架自动化了模型选择和评估过程。基于 F1 分数确定了最佳模型，并按照任务要求，在测试集上计算了该模型的 F1 得分并构建了混淆矩阵。

PyCaret 的使用使得能够以最少的编码工作高效地比较多种模型，并获得分类器性能的评估结果。