1. Цель

Целью данной лабораторной работы является обработка кардиологического датасета и построение бинарного классификатора по признаку Healthy_Status с применением AutoML-фреймворков. Основные задачи включают:

- загрузку и фильтрацию 5000 записей ЭКГ данных;
- формирование признакового пространства по выбранным параметрам;
- обучение моделей с помощью AutoML;
- оценка качества классификации по матрице ошибок и F1-мере.

1. 目的

本实验旨在处理心电图（ECG）数据集，并基于 Healthy_Status 特征构建一个二元分类器。主要目标包括：

- 加载并筛选 5000 条心电图数据；
- 构建由特定特征组成的特征空间；
- 使用 AutoML 框架训练分类模型；
- 通过混淆矩阵和 F1-分数评估模型性能。

2. Методы

1) Подготовка данных

Из датасета «модуль 3...» были выбраны 5000 строк и следующие столбцы:

2. 方法

1) 数据准备

从提供的“模块 3...”数据集中加载 5000 条记录，选取以下字段：

```
# загружаем полный датасет
raw_table_data = pd.read_csv('https://raw.githubusercontent.com/TAUforPython/BioMedAI/main/test_datasets/test_data_ECG.csv', nrows=5000)
# raw_table_data = pd.read_csv()
raw_table_data.head(10)
```

	subject_id	Count_subj	study_id	cart_id	Healthy_Status	eeg_time	eeg_date	report_0	report_1	report_2	...	filtering	rr_interval	p_onset	p_end	qrs_onset	qrs_end	t_end	p_axis	qrs_axis	t_axis
0	19557662	27	40000017	6848296	0	8:44 AM	27.06.2015	Sinus rhythm	Possible right atrial abnormality	NaN	...	60 Hz notch Baseline filter	659	40	128	170	258	518	81	77	79
1	18477137	93	40000029	6848296	0	9:54 AM	27.06.2015	Sinus rhythm	Possible right atrial abnormality	NaN	...	60 Hz notch Baseline filter	722	40	124	162	246	504	77	75	70
2	16598616	3	40000035	6376932	1	9:07 AM	28.06.2015	Sinus tachycardia	NaN	Normal ECG except for rate	...	60 Hz notch Baseline filter	600	40	130	162	244	474	79	72	77
3	16368287	7	40000079	6214760	1	5:14 PM	15.07.2015	Sinus rhythm	NaN	Normal ECG	...	60 Hz notch Baseline filter	659	40	146	180	254	538	79	66	69
4	18370366	2	40000084	6632385	0	1:52 PM	27.09.2015	Sinus rhythm	NaN	NaN	...	<not specified>	659	368	29999	504	590	868	84	80	77
5	15606157	55	40000089	6632385	0	2:29 PM	29.10.2013	Sinus rhythm	NaN	NaN	...	<not specified>	822	365	29999	499	592	852	26	46	30
6	12576058	43	40000115	6852956	1	12:54 PM	23.03.2016	Sinus rhythm	NaN	Normal ECG	...	60 Hz notch Baseline filter	952	40	146	198	282	598	24	80	20
7	14691089	1	40000143	6551957	0	10:01 AM	10.12.2016	Sinus rhythm	rSr'(V1) - probable normal variant	Low QRS voltages in precordial leads	...	60 Hz notch Baseline filter	923	40	140	188	278	594	26	86	13
8	14144725	7	40000144	6924910	0	7:24 AM	11.12.2011	Sinus rhythm with PAC(s).	NaN	Borderline ECG	...	60 Hz notch Baseline filter	952	40	180	196	294	610	59	-17	3
9	16089780	2	40000152	6919786	0	12:35 PM	13.12.2011	Sinus rhythm	Extensive T wave changes may be due to myocard...	NaN	...	60 Hz notch Baseline filter	1000	40	156	178	274	584	8	-11	19

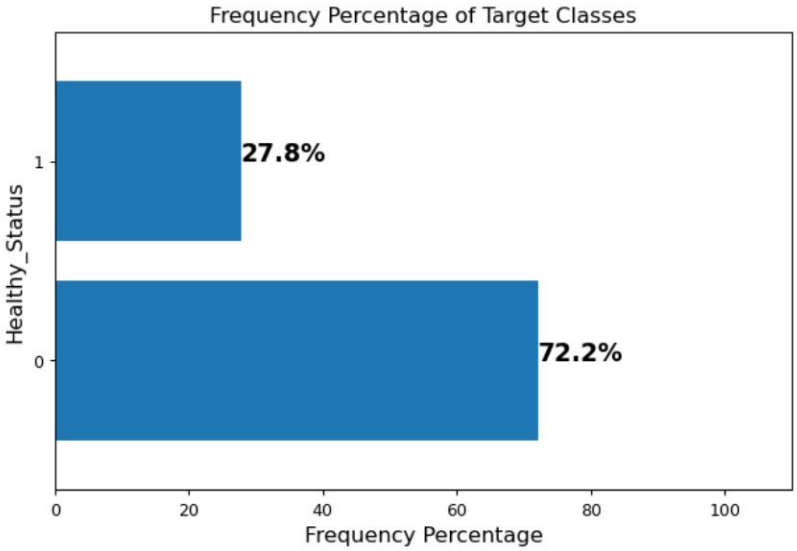
Очищение данных от выбросов осуществлялось с помощью логических условий:

应用逻辑筛选条件以清洗异常值（通过去除异常值、合并和清理文本报告数据，使用 Word2Vec 模型将文本转化为数值表示，并将其作为新的特征用于后续分析）：

```
# Убираем выбросы数据过滤，过滤掉所有在 columns_to_filter 列中具有大于 2000 的值的行，
# 筛选出 p_onset 小于 p_end 且 qrs_onset 小于 qrs_end 的行。这是为了去除数据中的异常值（如时间顺序错误等）
columns_to_filter = ['rr_interval', 'p_onset', 'p_end', 'qrs_onset', 'qrs_end', 't_end', 'p_axis', 'qrs_axis', 't_axis']
full_df_filtered = raw_table_data[(raw_table_data[columns_to_filter] < 2000).all(axis=1)]
full_df_filtered = full_df_filtered[(full_df_filtered['p_onset'] < full_df_filtered['p_end']) & (full_df_filtered['qrs_onset'] < full_df_filtered['qrs_end'])]

# Слепляем все текстовые отчеты в один большой文本数据预处理，
# 将 report_0 到 report_17 列中的文本数据合并成一个单一的文本列 report。每个报告通过空格连接。接着，去除字符串中的 nan 和多余的空格，确保文本格式清洁。
reports = [f'report_{x}' for x in range(18)]
full_df_filtered['report_0'] = full_df_filtered[reports].astype(str).agg(' '.join, axis=1)
full_df_filtered['report_0'] = full_df_filtered['report_0'].str.replace(r'\bnan\b', '', regex=True).str.replace(r'\s+', ' ', regex=True).str.strip()
full_df_filtered.rename(columns={'report_0': 'report'}, inplace=True)
reports_to_drop = [f'report_{x}' for x in range(1, 18)]
full_df_filtered = full_df_filtered.drop(reports_to_drop, axis=1)

# Фиксим имена столбцов 修正列名字；删除了 bandwidth 和 filtering 列（无关列）
full_df_filtered = full_df_filtered.rename(columns={'eeg_time ': 'eeg_time', 'eeg_date ': 'eeg_date'})
full_df_filtered = full_df_filtered.drop(columns = ['bandwidth', 'filtering'])

# Делаем колонку с таргетами крайней справа将 Healthy_Status 列移到数据框的最右侧
full_df_filtered = full_df_filtered[[col for col in full_df_filtered.columns if col != 'Healthy_Status'] + ['Healthy_Status']]

# Разбиваем столбец с текстовым отчетом на токены分词,将每个报告report的文本按空格进行分割，转换成词语列表
words = [text.split() for text in full_df_filtered['report']]

# Обучаем модель Word2Vec使用 Word2Vec 模型训练分词后的文本数据。这个模型会将每个词转换为一个向量。
w2v_model = Word2Vec(words)
```

Предварительно обработанные характеристики ЭКГ были классифицированы с использованием модели Гаусса Naive Bayes, после чего была проанализирована эффективность модели.

用 Gaussian Naive Bayes 模型对预处理后的 ECG 特征进行分类，并对模型进行性能分析。



2) Обучение модели AutoML

Изучены следующие AutoML-фреймворки:

- H2O AutoML: прост в применении, есть графический интерфейс, хорошо масштабируется.
- TPOT: генетический подход к построению моделей.
- AutoSklearn: использует байесовскую оптимизацию, высокое качество, требует больше времени.

Выбор: H2O AutoML как оптимальный по соотношению качества и простоты настройки.

2) AutoM 模型训练

对比以下 AutoML 框架：

- H2O AutoML：易用，支持图形界面，适合大数据；
- TPOT：基于遗传算法构建模型；
- AutoSklearn：基于贝叶斯优化，准确率高但耗时较长。

选择结果：H2O AutoML 平衡了效果与使用便捷性。

Здесь используется фреймворк AutoML от H2O для автоматического перебора нескольких комбинаций моделей (GLM, GBM, DRF, DeepLearning, StackedEnsemble и т. д.)

По умолчанию в качестве основы для оценки модели используются AUC и показатель F1.

使用 H2O 的 AutoML 框架自动尝试多种模型组合（GLM、GBM、DRF、DeepLearning、StackedEnsemble 等）。

默认采用 AUC、F1-score 为模型评估依据。

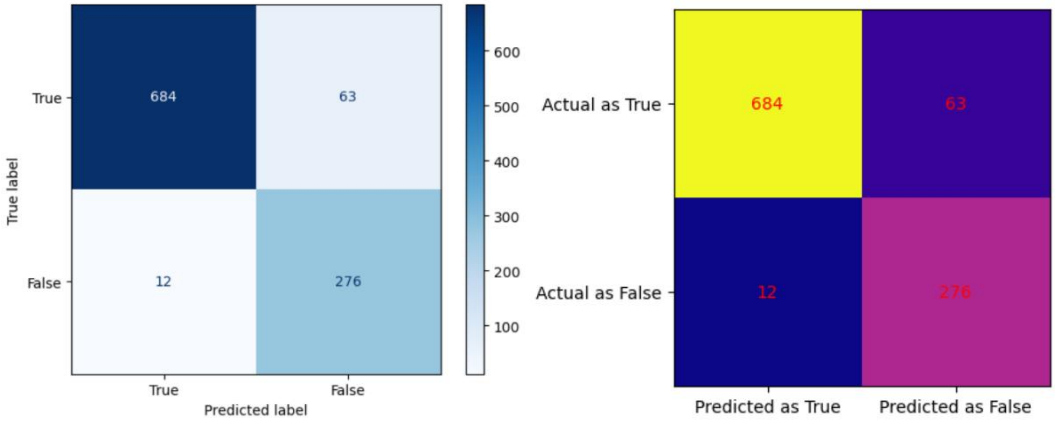
```
# 在 AutoML 中启用 XGBoost
from h2o.automl import H2OAutoML

automl = H2OAutoML(
    max_runtime_secs=60,
    seed=42,
    include_algos=["XGBoost", "GBM", "GLM", "DRF", "DeepLearning", "StackedEnsemble"],
    verbosity="info"
)
automl.train(x=x, y=y, training_frame=train_h2o)
```

3) Матрица путаницы и максимальные результаты F1

Матрица путаницы: оценивает эффективность модели классификации, показывая истинно положительные, ложно положительные, истинно отрицательные и ложно отрицательные результаты.

3)混淆矩阵与最大 F1 分数结果

混淆矩阵：评估分类模型的性能，显示真阳性、假阳性、真阴性和假阴性。



F1-мера: это гармоническое среднее значение точности и полноты, подходящее для оценки несбалансированных наборов данных.

F1 分数：综合考虑了精确率和召回率的调和平均数，适用于不平衡数据集的评估。

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

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

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

3. Сравнение моделей

3.模型对比

$$Precision = \frac{TP}{TP + FP} \approx 0.9766$$

$$Recall = \frac{TP}{TP + FN} \approx 1.0$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \approx 0.9881$$

Следующие экспериментальные данные показывают, что модель GBM_5 имеет наилучшие характеристики на тестовом наборе:
- В тестовом наборе только 3 отрицательных класса были классифицированы неправильно (почти идеальная классификация)
- Оценка F1 достигла 0,9881, что является одним из важнейших показателей в оценке
- AUC достигает 0,9998, и модель обладает сильной дискриминационной способностью.

以下实验数据说明， GBM_5 模型在测试集上评估性能最优：
- 在测试集上，仅有 3 个负类被误分（几乎完美分类）
- F1 分数达到 0.9881，这是评估中最重要的指标之一
- AUC 也高达 0.9998，模型具有极强区分能力

```
00:53:06.581: New leader: GBM_5_AutoML_2_

MSE: 0.0056893233587090985
RMSE: 0.07542760342678997
LogLoss: 0.01841294346956109
Mean Per-Class Error: 0.0038461538461538464
AUC: 0.9998153846153847
AUCPR: 0.9994308033611844
Gini: 0.9996307692307693

Confusion Matrix (Act/Pred) for max f1 @ threshold = 0.1346544244507371
       0    1    Error    Rate
-----  ---  ---  -------  -----------
0      387  3    0.0077   (3.0/390.0)
1      0    125  0        (0.0/125.0)
Total  387  128  0.0058   (3.0/515.0)

Maximum Metrics: Maximum metrics at their respective thresholds
metric                       threshold    value     idx
---------------------------  -----------  --------  -----
max f1                       0.134654     0.988142  115
```

Лучшая производительность модели (на тестовом наборе):

最优模型性能表现（在测试集上）：

Значение /指标	**Показателя/数值**	**Описание/说明**
AUC	0.9998	Модель обладает сильными возможностями классификации. 模型分类能力极强
F1-score	0.9881	Учитываются как точность, так и скорость отклика, почти идеальная. 准确率与召回率兼顾，几乎完美
Accuracy	≈ 0.994	Общая точность классификации высокая. 整体分类正确率高
LogLoss	0.01841	Вероятность выходных данных модели стабильна и достоверна.

Значение /指标	Показателя/数值	Описание/说明
		模型输出概率稳定且可信
Ошибка матрицы путаницы 混淆矩阵误差	Ложноположительный результат/假阳性 FP = 3 ложноотрицательный результат/假阴性 FN = 0	Почти все нездоровые люди были успешно идентифицированы (Полнота = 1,0) 非健康者几乎全部被成功识别（Recall = 1.0）

5. Заключение

В этом проекте используется фреймворк H2O AutoML для автоматического выбора и обучения нескольких моделей классификации для данных цифровых характеристик ЭКГ с целью решения задачи бинарной классификации (определения того, является ли человек «здоровым») и оценки эффективности на основе таких показателей, как матрица неточностей и показатель F1. Были получены следующие результаты:

- H2O AutoML может эффективно и автоматически искать и определять оптимальную модель классификации
- Характеристики ЭКГ имеют значительную дискриминационную силу для оценки состояния здоровья
- Окончательная модель GBM_5 имеет точность ≈ 99,4% и оценку F1 ≈ 0,988, что делает ее готовым к развертыванию высококачественным классификатором.

5.结论

本项目针对 ECG 数字特征数据，利用 H2O AutoML 框架自动选择和训练多个分类模型，解决二分类问题（判断个体是否为“健康”），并基于混淆矩阵、F1 分数等指标进行性能评估。得出以下结果：

- H2O AutoML 能够有效自动搜索并识别最优分类模型
- ECG 特征对于健康状态判断具有显著判别力
- 最终模型 GBM_5 准确率≈ 99.4%，F1-score ≈ 0.988，是一个可部署的高质量分类器

6. Ссылки на литературу

[1] L. van der Maaten, G. Hinton. Visualizing data using t-SNE, JMLR, 2008.
[2] I. T. Jolliffe, Principal Component Analysis, Springer, 2002.