

项目计划书

Echoes of Mood —— 基于海量歌词数据的情绪计算与智能推荐系统

1、项目摘要

在数字化音乐时代，海量曲库与用户个性化情感需求之间存在着巨大的匹配鸿沟。传统的协同过滤推荐算法往往基于历史行为数据，存在“冷启动”问题，且难以捕捉用户当下细腻、瞬时的心理状态。

本项目 Echoes of Mood (情绪共鸣) 旨在构建一个基于情绪计算的智能音乐推荐系统。项目利用开源的 ChineseLyrics 数据集，对 10 万+首中文歌曲进行全量 ETL (抽取、转换、加载) 清洗与挖掘。通过引入 BERT (Bidirectional Encoder Representations from Transformers) 预训练模型，将非结构化的歌词文本转化为高维语义向量，构建大规模情感索引库。

最终，系统将以 Web 应用的形式呈现，用户只需输入一段自然语言（如心情日记），系统即可在毫秒级时间内从海量曲库中检索出语义最契合的歌曲，实现“从文本到灵魂”的深度共鸣推荐。

2、项目背景与意义

2.1 行业痛点

- 推荐同质化：现有主流平台多推荐热门歌曲，导致“信息茧房”，大量优质的长尾 (Long-tail) 歌曲被埋没。
- 情感断层：标签式检索（如搜索“伤感”）粒度过粗，无法理解“热闹后的孤独”这类复杂情绪。

2.2 项目价值

- 大数据的深度应用：本项目不局限于小样本实验，而是直面 10W+ 级别的真实数据，实践从数据清洗到向量化检索的大数据全流程。
- AI 的人文关怀：探索人工智能在心理慰藉领域的应用，让算法具备“共情能力”，为用户提供具有温度的陪伴。

3、系统详细功能描述

本系统遵循高内聚、低耦合的设计原则，划分为三大核心功能子系统：数据治理与 ETL 子系统、情感计算与检索子系统、Web 交互与可视化子系统。

3.1 数据治理与 ETL 子系统

针对 ChineseLyrics 提供的 10W+ 原始非结构化数据，本子系统负责构建高质量的数据底座。

F1. 多源数据抽取

功能描述：系统具备流式读取能力，能够高效解析大规模 JSON 格式的原始歌词文件。

技术指标：支持 GB 级文本数据的快速加载，自动识别文件编码格式。

F2. 智能噪声清洗

功能描述：内置正则表达式引擎，自动剔除歌词文本中的非情感信息（如时间轴标签 [00:12.34]、编曲人信息、广告词等）。

规则引擎：

长度过滤：自动丢弃字数 < 30 的无效样本（如纯音乐占位符）。

去重处理：识别并合并同一首歌曲的不同版本（如 Live 版与录音室版），优先保留文本质量最高的版本。

F3. 结构化存储

功能描述：将清洗后的数据转化为结构化格式（CSV/Parquet），建立包含 Global_ID、Song_Name、Artist、Clean_Lyrics 的标准数据表，为上层应用提供统一视图。

3.2 情感计算与检索子系统

这是系统的“大脑”，负责理解自然语言并进行大规模相似度匹配。

F4. 文本向量化编码

功能描述：集成 BERT (bert-base-chinese) 预训练模型，作为文本到向量的编码器。

处理逻辑：

输入：任意长度的用户日记或歌词片段。

处理：截断/填充文本至 512 token，提取 Transformer 最后一层 [CLS] 位置的输出。

输出：生成 768 维的稠密实数向量（Dense Vector），该向量蕴含了文本的深层语义与情感倾向。

F5. 大规模向量索引构建

功能描述：针对全量 10W+ 歌曲生成的向量矩阵，构建高效的内存索引结构。

性能优化：采用 NumPy 矩阵运算优化，将所有歌曲向量加载至内存，避免频繁磁盘 I/O，确保检索过程的高吞吐量。

F6. 语义共鸣检索

功能描述：基于余弦相似度（Cosine Similarity）算法，计算“用户心境向量”与“全量歌词向量矩阵”的相似性得分。

推荐策略：

Top-K 筛选：实时返回相似度最高的 Top 3 歌曲。

语义阈值过滤：仅推荐相似度 > 0.6 的歌曲，确保推荐的准确性，宁缺毋滥。

3.3 Web 交互与可视化子系统

这是系统的“面孔”，负责为用户提供治愈、流畅的沉浸式体验。

F7. 沉浸式情感输入

功能描述：提供极简的文本输入界面，支持动态背景（根据时间或随机算法变化）。

交互细节：用户输入时伴有柔和的呼吸光效，鼓励用户通过文字宣泄情绪。

F8. 智能推荐卡片

功能描述：以卡片形式展示推荐结果。

内容包含：

基础信息：歌名、歌手（突出显示 Eason、JJ 等知名歌手，也不放过冷门佳作）。

情感高亮：系统自动截取歌词中与用户输入最“共鸣”的片段进行展示，让用户瞬间明白“为什么推荐这首歌”。

试听链接：集成外部音乐平台（网易云/QQ 音乐）API，支持一键跳转播放。

F9. 响应式反馈

功能描述：在后端进行复杂向量计算时，前端展示优雅的 Loading 动画（如声波律动），缓解用户等待焦虑，确保毫秒级的交互体感。

4、系统框架

4.1 数据层

4.1.1 数据源

ChineseLyrics 数据集：这是本项目的核心数据源，包含了超过 10 万首中文歌曲的歌词。数据集采用 JSON 格式，包含多个字段，如歌曲标题、歌手信息、歌词内容等。该数据集为情绪计算和音乐推荐提供了基础数据支持。

4.1.2 ETL 流程

为了确保数据的有效性与可用性，采用了 ETL（提取、转换、加载）流程：

提取（Extract）：从多个 JSON 文件中流式读取数据。每个文件包含数千首歌的歌词、歌手信息等。

转换（Transform）：

正则清洗：通过正则表达式去除无用的元数据（如时间戳和作词/作曲人信息），保留歌词的核心内容。

长度过滤：剔除字数少于 30 的短歌词文本，避免噪音数据。

去重处理：处理同一首歌的不同版本（如不同演唱者的版本），确保每首歌曲只出现一次。

加载（Load）：经过清洗和转换的数据最终被加载到 CSV 文件和 NumPy 矩阵中，以供后续的数据处理与向量化。

4.2 算法层

4.2.1 模型与特征提取

模型选型: 项目使用了 BERT (中文版), 即 bert-base-chinese, 这一预训练模型可以理解和捕捉中文歌词中的深层语义。

特征提取:

通过 BERT 模型对清洗后的歌词数据进行处理, 提取出每首歌的“情绪指纹”。

使用 BERT 模型输出的[CLS]位置向量, 作为每首歌的情绪表示。该向量可以有效地捕捉歌词文本的情感和语义特征。

4.2.2 向量化与检索

向量化过程:

通过批量处理, 利用 BERT 将所有歌词转换为 768 维的向量 (每首歌一个向量), 存储为 NumPy 矩阵 (embeddings.npy)。

检索引擎:

使用 余弦相似度 (Cosine Similarity) 来计算用户输入的文本向量与歌词向量之间的相似度。余弦相似度能够衡量两个向量之间的角度差异, 从而判断它们在情感上的相似程度。

采用内存矩阵计算方式, 在内存中并行计算用户向量与 10 万首歌向量的相似度, 确保检索响应延迟控制在 200ms 以内。

4.3 应用层

4.3.1 后端服务

Flask 框架: 使用 Flask 作为后端服务器框架, 负责处理前端请求、调用情感计算模块, 并返回最相关的歌曲推荐。

API 设计:

输入 API: 用户输入一段自然语言文本 (如心情日记), 该 API 接收并将文本传递给情感分析模块。

推荐 API: 基于输入的文本和歌曲情感向量库, 返回与用户情感匹配度最高的歌曲列表。该 API 会计算用户输入与各个歌曲的情感向量的相似度, 排序并返回最契合的歌曲。

4.3.2 前端交互

前端技术栈: 使用 HTML5、CSS3 (Flex 布局)、JavaScript (AJAX) 进行前端开发, 确保系统具有简洁、直观的用户界面。

界面设计:

极简风格: 系统采用治愈系的极简设计风格, 注重“呼吸感”和“沉浸感”, 以增强用户体验。设计强调页面的轻盈与舒适感, 帮助用户在使用过程中获得情感上的放松与共鸣。

交互流程:

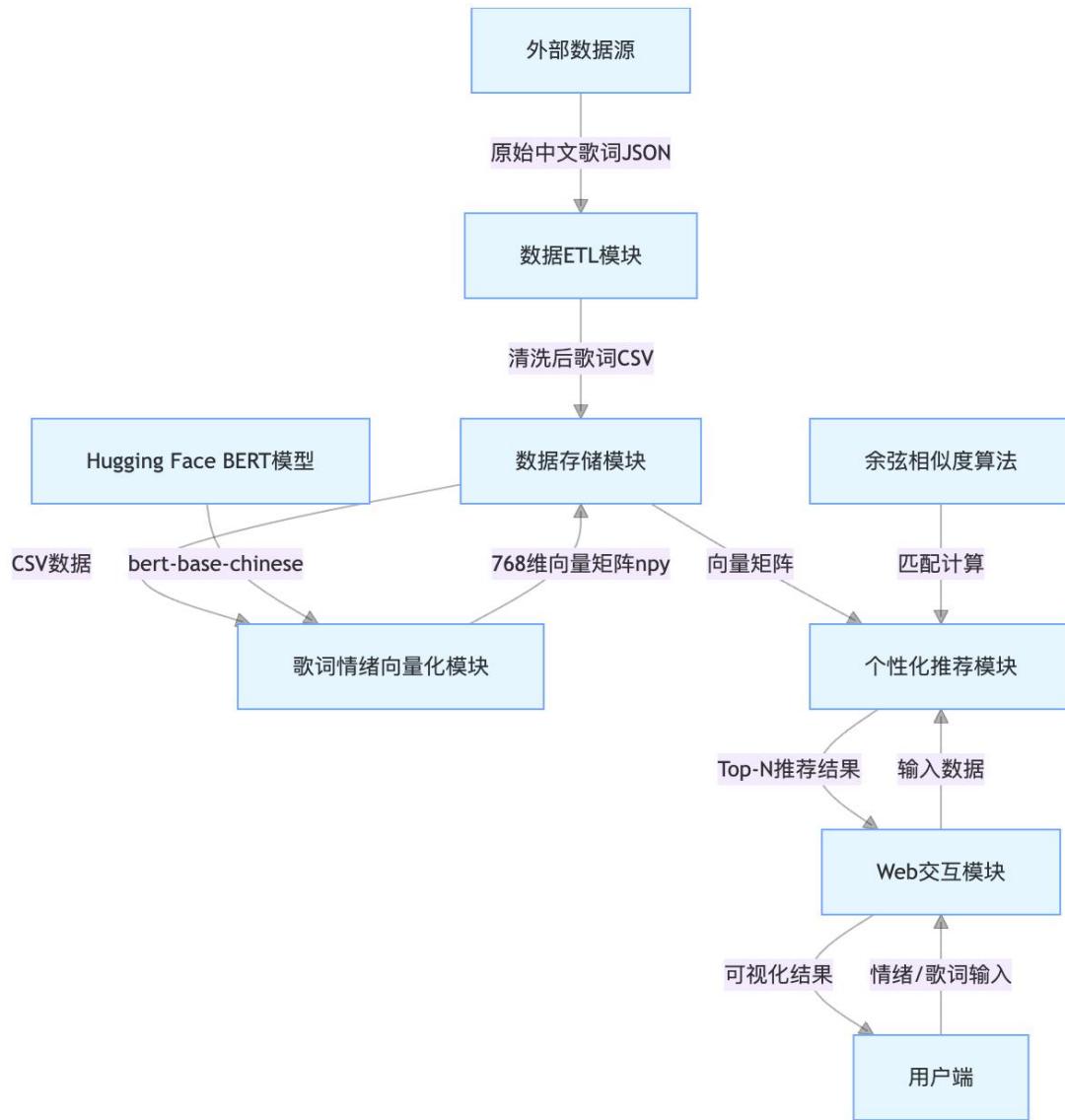
用户在输入框中输入心情描述或其他自然语言文本。

前端通过 AJAX 异步请求后端 API, 将用户的输入发送至服务器。

后端接收并处理请求后，返回匹配度最高的歌曲推荐，前端界面展示结果。

5、关键模块与 API

5.1 模块整体架构图

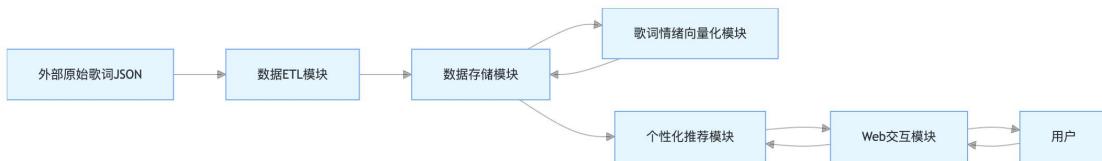


- (1) 数据层：外部数据源提供原始歌词，经 ETL 清洗后存入数据存储模块，后续模块提供标准化数据；
- (2) 算法层：向量化模块依赖 BERT 模型生成“情绪指纹”向量，推荐模块基于余弦相似度实现精准匹配；
- (3) 应用层：Web 交互模块承接用户输入与结果展示，实现“输入 - 计算 - 输出”的端到端交互。

5.2 关键模块设计

5.2.1 关键模块概览

模块名称	核心功能	核心输入输出
数据 ETL 模块	清洗原始歌词数据, 生成标准化 CSV	输入: 原始 JSON 歌词 → 输出: clean_lyrics.csv
歌词情绪向量化模块	提取歌词情绪特征, 生成向量矩阵	输入: CSV 歌词 → 输出: embeddings.npy (768 维向量)
个性化推荐模块	基于情绪 / 歌词匹配, 输出 Top-N 推荐	输入: 情绪标签 / 歌词片段 → 输出: 歌曲推荐列表
Web 交互模块	承接用户交互, 串联前后端流程	输入: 用户操作 → 输出: 可视化推荐结果
数据存储模块	分类存储核心数据, 提供读写支持	输入: CSV / 向量文件 → 输出: 数据读取接口



5.2.2 各模块详情

5.2.2.1 数据 ETL 模块

(1) 核心职责:

提取多来源的原始歌词 JSON 数据;
清洗时间戳、作者标注等无效格式;
将处理后的数据加载为标准化的 CSV 文件 (clean_lyrics.csv) 。

(2) 核心流程:

提取阶段: 读取 5 个 JSON 数据源, 支持批量处理, 单批次最大处理 10 万条原始数据;
数据校验: 过滤空值、重复数据;
清洗阶段: 删除 [00:12.00] 格式时间戳、“作词:xxx” 等标注 (通过正则表达式自动识别匹配);
内容筛选: 保留字符数 ≥ 5 且 < 30 的有效歌词;
加载阶段: 按 “歌词 ID - 歌曲名 - 歌词内容” 格式, 以 UTF-8 编码存储为 CSV 文件。

(3) 依赖项:

工具: JSON 解析库、Pandas (数据处理) 、NumPy;
模块: 数据存储模块 (接收 CSV) 。

(4) 关键指标:

数据清洗率 $\geq 95\%$ (有效数据 / 原始数据) ;
10 万条数据 ETL 总耗时 ≤ 30 分钟。

5.2.2.2 歌词情绪向量化模块

(1) 核心职责:

加载 bert-base-chinese 模型;
批量将 CSV 歌词转化为 768 维向量;
支持单条歌词实时向量化;

输出 embeddings.npy 向量矩阵。

(2) 核心流程:

输入数据: 批量处理接收 clean_lyrics.csv, 单条处理接收歌词文本;
模型初始化: 加载 BERT 预训练权重与 Tokenizer;
文本预处理: 分词 → 添加 [CLS] / [SEP] 标记 → 编码转换;
特征提取: 取 [CLS] token 输出特征向量;
向量生成: 批量生成 (N, 768) 维度矩阵 (N 为歌词数量), 单条生成 (1, 768) 数组;
结果存储: 批量结果存储为 embeddings.npy, 单条结果以 JSON 格式返回。

(3) 依赖项:

工具: Hugging Face Transformers、NumPy;
模块: 数据 ETL 模块 (输入 CSV) 、数据存储模块 (存储 npy) 。

(4) 关键指标:

单条向量化响应时间 < 200ms;
10 万条数据批量向量化耗时 ≤ 2 小时;
向量维度统一为 768 维。

5.2.2.3 个性化推荐模块

(1) 核心职责:

接收用户输入 (情绪标签 / 歌词片段);
输入向量化 (情绪→预设向量; 歌词→调用向量化 API);
余弦相似度计算匹配向量;
按相似度降序输出 Top-N 推荐结果。

(2) 核心流程:

用户输入: 支持 “情绪标签” (开心 / 伤感 / 平静 / 激昂) 或 “歌词片段” 两种输入类型;
输入处理: 情绪标签映射为预设向量 (由 1000 条对应情绪歌词向量平均得到), 歌词片段调用单条向量化 API;
相似度计算: 通过余弦相似度算法对比 embeddings.npy 中所有向量;
结果过滤: 保留相似度 ≥ 设定阈值的向量;

排序输出: 按相似度得分降序取 Top-N 结果, 返回歌曲名、歌词片段、
匹配得分。

(3) 依赖项:

工具: Scikit-learn (余弦相似度计算) 、NumPy;
模块: 歌词情绪向量化模块 (向量来源) 、Web 交互模块 (接收输入 / 返回结果) 。

(4) 关键指标:

推荐结果匹配准确率 $\geq 80\%$ (用户满意度调研) ;
Top-10 推荐响应时间 < 1 秒;
推荐结果中热门歌曲占比 $\leq 30\%$ 。

5.2.2.4 Web 交互模块

(1) 核心职责:

基于 Flask 搭建后端服务;
提供前端页面 (输入区 + 结果区 + 历史记录) ;
接收用户输入并转发至推荐模块;
格式化推荐结果供前端渲染。

(2) 核心流程:

用户访问: 通过浏览器 / 移动端访问系统;
首页加载: 返回包含 CSS/JS 资源的 HTML 页面;
前端交互: 用户选择输入类型、填写内容、设置 Top-N 数量 (1-20) ;
AJAX 请求: 通过异步请求调用后端接口, 避免页面刷新;
后端处理: 接收请求并转发至个性化推荐模块;
结果返回: 格式化推荐结果 (含歌曲封面图路径、相似度得分) ;
前端渲染: 以卡片式布局展示推荐列表。

(3) 依赖项:

工具: Flask、HTML5、CSS3 (Flex 布局) 、JavaScript (AJAX) ;
模块: 个性化推荐模块 (推荐结果来源) 。

(4) 关键指标:

页面加载时间 < 2 秒;
跨浏览器兼容性 (支持 Chrome/Firefox/Edge) ;
异步请求响应流畅。

5.2.2.5 数据存储模块

(1) 核心职责:

分类存储系统核心数据;
提供数据读取接口 (各模块调用) ;
支持数据备份与版本管理。

(2) 核心流程:

数据输入: 接收 ETL 模块输出的 CSV 文件、向量化模块输出的 npy 文件、推荐模块生成的日志数据;
存储分类: 结构化数据 (CSV) 采用关系型存储, 非结构化数据 (npy、模型权重) 采用文件存储;
数据索引: 建立歌词 ID 与向量的关联索引, 加速推荐查询;
接口提供: 提供数据读取接口, 按模块权限返回对应数据;
数据安全: 定期备份数据 (≥ 1 次 / 天), 避免数据丢失。

(3) 依赖项:

工具: SQLite (可选, 结构化存储) 、文件系统;
模块: 所有业务模块 (数据读写) 。

(4) 关键指标:

数据读取响应时间 < 100ms;
数据备份频率 ≥ 1 次 / 天;
数据存储准确率 100% (无丢失 / 错配) 。

5.3 API 设计

5.3.1 API 设计规范

- (1) 路径格式: /api/v{版本号}/{模块名}/{功能名} (如 /api/v1/etl/extract) ;
- (2) 请求方式: GET (查询 / 获取资源, 无数据修改) 、POST (创建 / 提交数据, 涉及数据处理或状态变更) ;
- (3) 参数格式:
 - 1) Query 参数: 适用于简单查询场景 (如单条歌词向量化、推荐结果查询), 直接拼接在 URL 后, 格式为 "?key1=value1&key2=value2" ;
 - 2) Body 参数: 适用于复杂数据提交 (如批量 ETL、推荐请求), 采用 JSON 格式, 确保参数结构清晰、可扩展;
- (4) 返回格式: 统一 JSON 格式, 包含 3 个核心字段:
 - 1) status: 请求状态 (success/error) , 用于快速判断请求是否成功;
 - 2) message: 说明信息 (成功时返回操作结果描述, 失败时返回简要错误说明) 。
 - 3) data: 业务数据 (成功时返回核心结果, 失败时返回 error_code 和 error_msg) 。
- (5) 错误处理: 采用“错误码 + 错误详情”机制, 错误码按模块分区 (1000-1999 对应 ETL 模块、2000-2999 对应向量化模块等), 便于快速定位问题;
- (6) 性能要求: 所有 API 响应时间 ≤ 1 秒 (单条向量化 ≤ 200ms、推荐查询 ≤ 1 秒) , 满足用户实时交互需求。

5.3.2 API 概览

模块名称	API 路径	请求方式	核心功能	适用场景
数据 ETL 模块	/api/v1/etl/extract	POST	批量提取多来源原始 JSON 歌词数据	系统初始化数据准备
数据 ETL 模块	/api/v1/etl/transform	POST	按规则清洗原始数据, 过滤无效内容	提取后的数据标准化处理
数据 ETL 模块	/api/v1/etl/load	POST	将清洗后数据加载为标准化 CSV 文件	清洗后数据存储与调用
向量化模块	/api/v1/embedding/batch	POST	批量 CSV 歌词转化为 768 维情绪向量矩阵	系统初始化向量生成
向量化模块	/api/v1/embedding/single	GET	实时生成单条歌词的 768 维情绪向量	用户输入歌词推荐的向量化
个性化推荐模块	/api/v1/recommend/mood	POST	根据情绪标签返回 Top-N 个性化歌曲推荐	用户明确情绪需求的推荐
个性化推荐模块	/api/v1/recommend/lyric	POST	根据歌词片段返回风格相似的歌曲推荐	用户喜欢歌词的相似推荐
Web 交互模块	/api/v1/web/index	GET	返回系统首页 HTML 页面及关联静态资源	用户访问系统入口
Web 交互模块	/api/v1/web/user/input	POST	接收用户推荐请求, 生成唯一查询 ID	前端提交用户推荐需求
Web 交互模块	/api/v1/web/recommend/result	GET	根据查询 ID 返回格式化的推荐结果	前端查询并渲染推荐结果

5.3.3 各 API 详细说明

5.3.3.1 数据 ETL 模块 API

(1) /api/v1/etl/extract

请求方式: POST

参数类型: Body

核心参数: json_paths (原始 JSON 文件路径列表, 必填)、source_tag (数据源标识, 可选)、encoding (文件编码, 可选)

功能说明: 批量读取指定路径的原始 JSON 歌词数据, 过滤空文件和无效 JSON 格式, 统计提取数据量并生成临时数据 ID, 供后续清洗操作调用。支持一次提取多个文件, 提升数据处理效率。

返回结果: 返回提取数据条数、耗时、临时数据 ID (data_id) 等信息, data_id 有效期 24 小时。

错误处理: JSON 文件不存在 (错误码 1001)、编码解析失败 (错误码 1002)、JSON 格式无效 (错误码 1005)。

(2) /api/v1/etl/transform

请求方式: POST

参数类型: Body

核心参数: data_id (提取阶段生成的临时数据 ID, 必填)、filter_rules (清洗规则列表, 必填)

功能说明: 根据指定规则 (移除时间戳、作者标注、过滤短歌词等) 清洗原始数据, 剔除空歌词、无效格式内容, 生成清洗后数据 ID (clean_data_id), 并统计无效数据原因。

返回结果: 返回有效 / 无效数据条数、清洗耗时、clean_data_id、无效数据原因统计。

错误处理: 数据 ID 不存在 (错误码 1003)、过滤规则无效 (错误码 1006)。

(3) /api/v1/etl/load

请求方式: POST

参数类型: Body

核心参数: clean_data_id (清洗阶段生成的 ID, 必填)、target_path (CSV 存储路径, 必填)、file_name (CSV 文件名, 可选)

功能说明: 将清洗后的有效数据按 “歌词 ID - 歌曲名 - 歌词内容” 格式, 以 UTF-8 编码加载为 CSV 文件, 存储到指定路径, 为向量化模块提供标准化数据。

返回结果: 返回 CSV 文件完整路径、文件大小、列名等信息。

错误处理: 目标路径无写入权限 (错误码 1004)、清洗数据 ID 过期 (错误码 1007)。

5.3.3.2 歌词情绪向量化模块 API

(1) /api/v1/embedding/batch

请求方式: POST

参数类型: Body

核心参数: csv_path (ETL 生成的 CSV 路径, 必填)、bert_model (BERT

模型版本, 可选)、batch_size (批次大小, 可选)、npy_path (向量存储路径, 必填)

功能说明: 系统初始化阶段核心接口, 读取 CSV 中所有有效歌词, 通过 bert-base-chinese 模型批量生成 768 维情绪向量, 存储为.npy 格式文件 (支持快速读取计算), 批次大小可根据服务器性能调整。

返回结果: 返回向量矩阵文件路径、向量化歌词条数、平均耗时、矩阵维度等信息。

错误处理: CSV 文件不存在 (错误码 2001)、模型加载失败 (错误码 2002)、批次大小无效 (错误码 2004)。

(2) /api/v1/embedding/single

请求方式: GET

参数类型: Query

核心参数: lyric_text (单条歌词文本, 必填)、bert_model (BERT 模型版本, 可选)

功能说明: 用于用户实时输入歌词推荐场景, 接收用户输入的歌词文本 (≤ 50 字符, 过长自动截取), 实时调用 BERT 模型生成 768 维向量, 响应时间 $\leq 200\text{ms}$ 。

返回结果: 返回 768 维向量数组、响应耗时、输入歌词文本等信息。

错误处理: 歌词文本为空 (错误码 2003)、文本过长 (错误码 2005)。

5.3.3.3 个性化推荐模块 API

(1) /api/v1/recommend/mood

请求方式: POST

参数类型: Body

核心参数: mood_tag (情绪标签, 必填, 支持 “开心 / 伤感 / 平静 / 激昂”)、top_n (推荐数量, 必填, 1-20)、similarity_threshold (相似度阈值, 可选, 0.3-0.9)、npy_path (向量矩阵路径, 必填)

功能说明: 针对用户明确情绪需求, 匹配系统内置情绪预设向量 (由 1000 条对应情绪歌词向量平均生成), 通过余弦相似度算法筛选全量歌词向量中匹配度 \geq 阈值的结果, 按得分降序返回 Top-N 歌曲, 热门歌曲占比 $\leq 30\%$ 。

返回结果: 返回推荐歌曲列表 (含歌曲名、歌词片段、相似度得分等)、请求参数回显、计算耗时。

错误处理: 情绪标签无效 (错误码 3001)、向量矩阵文件不存在 (错误码 3002)、top_n / 阈值无效 (错误码 3004/3005)。

(2) /api/v1/recommend/lyric

请求方式: POST

参数类型: Body

核心参数: input_lyric (用户输入歌词片段, 必填, 5-100 字符)、top_n (推荐数量, 必填, 1-20)、similarity_threshold (相似度阈值, 可选)、npy_path (向量矩阵路径, 必填)

功能说明: 针对用户喜欢某句歌词的场景, 先将输入歌词片段向量化, 再

对比全量歌词向量矩阵，筛选相似度 \geq 阈值的结果，返回风格相似的Top-N歌曲，匹配核心为情绪风格而非关键词。

返回结果：返回推荐歌曲列表、向量化耗时、相似度计算耗时。

错误处理：无匹配结果（错误码 3003）、歌词片段过短（错误码 3006）、向量矩阵文件不存在（错误码 3002）。

5.3.3.4 Web 交互模块 API

(1) /api/v1/web/index

请求方式：GET

参数类型：无

功能说明：前端入口接口，用户访问系统时，后端返回完整首页 HTML 结构（含 Logo、输入区、结果区、底部信息）及关联的 CSS/JS 静态资源，支持跨浏览器访问，页面加载时间 ≤ 2 秒。

返回结果：完整的 HTML 页面内容。

错误处理：页面资源缺失（错误码 4001）、服务器内部错误（错误码 4007）。

(2) /api/v1/web/user/input

请求方式：POST

参数类型：Body

核心参数：input_type（输入类型，必填，“mood” / “lyric”）、input_content（输入内容，必填）、top_n（推荐数量，必填，1-20）、similarity_threshold（相似度阈值，可选）

功能说明：接收前端用户的推荐请求，统一参数格式后转发至个性化推荐模块异步处理，返回唯一请求 ID（request_id），便于前端后续查询结果，避免页面长时间阻塞。

返回结果：返回 request_id、请求提交时间、预计等待时间。

错误处理：输入类型无效（错误码 4002）、输入内容无效（错误码 4008）。

(3) /api/v1/web/recommend/result

请求方式：GET

参数类型：Query

核心参数：request_id（推荐请求 ID，必填）

功能说明：前端查询推荐结果的核心接口，根据 request_id 查询任务状态：未完成则返回处理中提示，完成则返回格式化结果（含歌曲封面图路径），供前端卡片式渲染。

返回结果：返回推荐歌曲列表（含封面图路径）、request_id 回显。

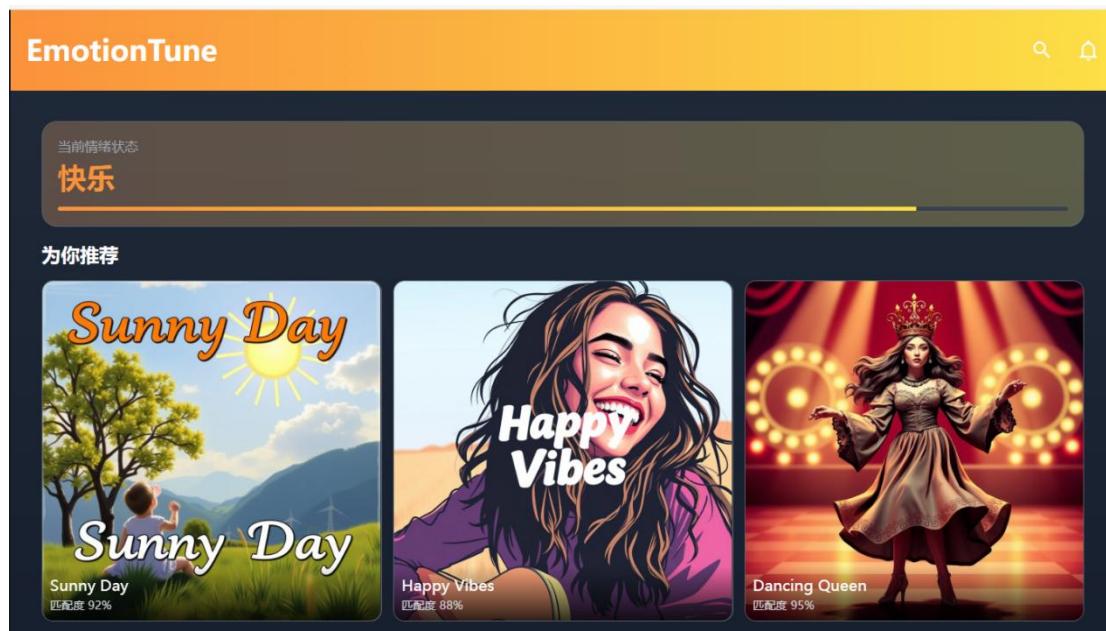
错误处理：请求 ID 无效 / 过期（错误码 4003）、任务处理中（错误码 4009）、任务处理失败（错误码 4010）。

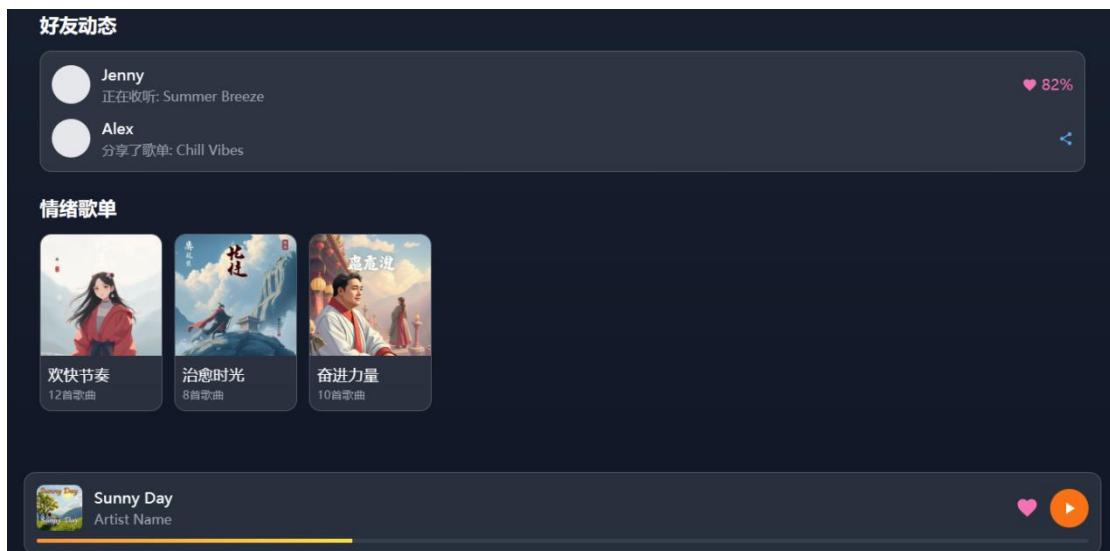
5.4 错误汇总表

错误码范围	所属模块	错误码	错误描述	解决方案
1000-1999	数据 ETL 模块	1001	JSON 文件不存在	检查 JSON 文件路径是否正确，确保文件已上传
1000-1999	数据 ETL 模块	1002	UTF-8 编码解析失败	转换文件编码为 UTF-8，或指定正确编码格式
1000-1999	数据 ETL 模块	1003	清洗数据 ID 不存在	先执行 /api/v1/etl/extract 获取有效数据 ID

1000-1999	数据 ETL 模块	1004	目标路径无写入权限	为目标文件夹分配写入权限 (如 chmod 755)
1000-1999	数据 ETL 模块	1005	JSON 格式无效	修复原始文件的 JSON 结构, 确保格式正确
1000-1999	数据 ETL 模块	1006	过滤规则无效	按 支 持 的 规 则 修 改 filter_rules 参 数 (remove_timestamp、remove_author、min_length=N)
1000-1999	数据 ETL 模块	1007	清洗数据 ID 过期	重新执行 /api/v1/etl/transform 获取新的 clean_data_id
2000-2999	向量化模块	2001	CSV 文件不存在	检查 CSV 路径, 确保 ETL 模块已生成 clean_lyrics.csv
2000-2999	向量化模块	2002	BERT 模型加载失败	检查网络 (自动下载) 或手动放置模型文件到指定目录
2000-2999	向量化模块	2003	歌词文本为空	输入有效歌词文本 (字符数≥1)
2000-2999	向量化模块	2004	批次大小无效	将 batch_size 调整为 16-64 之间的数值
2000-2999	向量化模块	2005	文本过长	歌词文本长度≤50 字符, 系统自动截取前 50 字符
3000-3999	个性化推荐模块	3001	情绪标签无效	选择支持的情绪标签 (开心 / 伤感 / 平静 / 激昂)
3000-3999	个性化推荐模块	3002	向量矩阵 npy 文件不存在	先执行 /api/v1/embedding/batch 生成 npy 文件
3000-3999	个性化推荐模块	3003	无匹配推荐结果	降低相似度阈值 (如从 0.7 改为 0.5) 或修改输入内容
3000-3999	个性化推荐模块	3004	top_n 无效	将 top_n 调整为 1-20 之间的数值
3000-3999	个性化推荐模块	3005	阈值无效	将 similarity_threshold 调整为 0.3-0.9 之间的数值
3000-3999	个性化推荐模块	3006	歌词片段过短	输入至少 5 个字符的歌词片段
4000-4999	Web 交互模块	4001	首页资源缺失	检查静态文件 (CSS/JS) 路径, 确保文件存在
4000-4999	Web 交互模块	4002	输入类型无效	将 input_type 改为 "mood" 或 "lyric"
4000-4999	Web 交互模块	4003	请求 ID 无效 / 过期	重新提交用户输入获取新 request_id
4000-4999	Web 交互模块	4007	服务器内部错误	联系管理员排查后端服务状态
4000-4999	Web 交互模块	4008	输入内容无效	按输入类型要求补充或修改 input_content
4000-4999	Web 交互模块	4009	任务处理中	前端继续定时轮询查询结果
4000-4999	Web 交互模块	4010	任务处理失败	重新提交推荐请求

6、UI 设计





首页 发现 播放 社区 我的 AI助手

现在播放

Sunny Day

Artist Name
Album Name

92% 情绪匹配

0:00 3:45

点播

分享

歌单

AI情绪分析

该歌曲具有欢快、充满活力的特征，完美匹配你当前的快乐情绪。建议结合此曲与其他舒缓音乐，打造全天最佳心情配置。

个人中心

用户昵称
Lv.8 | 🎵 音乐达人

128 已听曲数

12 创建歌单

45 粉丝数

播放历史

Sunny Day
Artist Name • 今天 14:30

Happy Vibes
Artist Name • 今天 12:45

我的收藏

32 已点赞

8 我的歌单

15 收藏歌单

6 分享统计

偏好设置

深色/浅色模式
当前: 深色

高对比度模式
无障碍功能

数据和隐私
管理个人数据

发现

社区

15

发现

社区



社区页面

敬请期待...

搜索

搜索歌曲、歌手...



搜索结果

输入关键词搜索...