全国省级居民消费指数(CPI)数据分析系统技术文档

1. 数据范围

1.1 数据内容

* 核心指标：省级消费者价格指数(CPI)
* 覆盖区域：全国31个省级行政区
* 时间跨度：2010年1月-2024年5月
* 数据粒度：月度统计

1.2 地理范围

| 区域 | 包含省份 |
| --- | --- |
| 华北 | 北京市、天津市、河北省、山西省、内蒙古自治区 |
| 东北 | 辽宁省、吉林省、黑龙江省 |
| 华东 | 上海市、江苏省、浙江省、安徽省、福建省、江西省、山东省 |
| 华中 | 河南省、湖北省、湖南省 |
| 华南 | 广东省、广西壮族自治区、海南省 |
| 西南 | 重庆市、四川省、贵州省、云南省、西藏自治区 |
| 西北 | 陕西省、甘肃省、青海省、宁夏回族自治区、新疆维吾尔自治区 |

1.3 时间范围

* 起始：2010年1月
* 结束：2024年5月
* 格式：YYYY-MM

2. 数据采集方案

2.1 数据来源

* 国家统计局公开数据
* 地区编码参照国家统计局标准

2.2 数据持久化

* 存储格式：MySQL关系型数据库
* 加载机制：
  + 首次运行创建数据库表结构并插入模拟数据
  + 后续运行直接从数据库查询数据
* 容错机制：数据库连接失败时记录错误日志

2.3 数据更新策略

| 策略类型 | 实现方式 |
| --- | --- |
| 静态数据 | 预生成模拟数据并插入数据库 |
| 动态扩展 | 1. 接入统计局API 2. 增量更新 3. 版本控制 |
| 容错机制 | 数据库连接失败时自动使用空数据集 |

2.4 数据初始化

系统启动时执行以下流程：

1. 检查MySQL数据库连接
2. 创建provincial\_cpi表（如果不存在）
3. 检查表中数据量
4. 如果数据为空，生成模拟数据并插入数据库
5. 从数据库加载CPI数据到内存

3. 数据存储结构

3.1 原始数据(MySQL表结构)

**表名：provincial\_cpi**

**字段说明：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| id | INT | 自增主键 |
| province | VARCHAR(50) | 省份名称 |
| date | VARCHAR(10) | 年月(YYYY-MM) |
| cpi | DECIMAL(5,2) | CPI指数(基期100) |

3.2 分析数据结构

**python**

{

"analysis\_img": "base64图表",

"wordcloud\_img": "base64词云",

"statistics": { /\* 统计指标 \*/ },

"raw\_data": [ /\* 原始数据子集 \*/ ]

}

**统计指标：**

| 指标 | 类型 | 说明 |
| --- | --- | --- |
| start\_date | 字符串 | 起始年月 |
| end\_date | 字符串 | 结束年月 |
| data\_points | 整数 | 数据点数量 |
| avg\_cpi | 浮点数 | 平均CPI |
| min\_cpi | 浮点数 | 最低CPI |
| max\_cpi | 浮点数 | 最高CPI |
| current\_cpi | 浮点数 | 最新CPI |
| std\_dev | 浮点数 | 标准差 |
| last\_change | 浮点数 | 最新环比变化 |

4. 分析目标与方法

4.1 分析目标

1. 时空规律：揭示省级CPI变化趋势与区域差异
2. 通胀监测：识别高通胀(CPI≥103)和通缩(CPI≤100)
3. 季节特征：分析月度周期性波动
4. 区域对比：比较七大地理区域CPI水平

4.2 技术方法

**核心分析流程**

text

数据加载 → 日期解析 → 特征提取 → 统计分析 → 可视化 → 词云生成

**关键技术实现**

1. 时间序列分析

python

*# 计算移动平均线*

df['12m\_ma'] = df['cpi'].rolling(window=12).mean()

1. 区域聚合分析

python

region\_avg = df.groupby('region')['cpi'].mean()

1. 统计指标计算

python

stats = {

'last\_change': df['cpi'].iloc[-1] - df['cpi'].iloc[-2]

}

1. 3关键词词云

python

*# CPI关键词映射*

if cpi >= 103: keywords = ['高通胀','物价上涨']

elif cpi <= 100: keywords = ['通缩','经济低迷']

4.4 可视化方案

**分析矩阵(2×2)：**

| 位置 | 省份分析 | 全国分析 |
| --- | --- | --- |
| 左上图 | CPI趋势+移动平均 | 各省平均CPI(条形图) |
| 右上图 | 年度平均CPI(柱状图) | 最新月份排名 |
| 左下图 | 月度季节模式(折线图) | 区域对比(柱状图) |
| 右下图 | 近期变化(柱状图) | 全国分布(直方图) |

4.5 分析输出

| 输出类型 | 内容说明 |
| --- | --- |
| 可视化图表 | 4图表分析矩阵(Base64) |
| 统计指标 | 8项核心指标(JSON) |
| 关键词词云 | 经济状态描述词云(Base64) |
| 原始数据 | 按时间倒序排列的子集 |

5. 数据库配置

**系统使用以下MySQL配置：**

text

[数据库]

host = localhost

user = root

password = 111111

database = cpi\_data

port = 3306

**表结构SQL：**

sql

CREATE TABLE provincial\_cpi (

id INT AUTO\_INCREMENT PRIMARY KEY,

province VARCHAR(50) NOT NULL,

date VARCHAR(10) NOT NULL,

cpi DECIMAL(5,2) NOT NULL,

UNIQUE KEY unique\_province\_date (province, date)

);

6. 系统架构

数据生成 → MySQL存储 → Flask API → 前端展示

**API服务包含：**

- 数据查询：原始数据获取

- 数据分析：多维分析结果

- 省份列表：区域信息获取

7. 技术栈

| 模块 | 技术 |
| --- | --- |
| 后端框架 | Flask |
| 数据处理 | Pandas, NumPy |
| 可视化 | Matplotlib |
| 词云生成 | WordCloud |
| 数据库 | MySQL (pymysql驱动) |
| 日志管理 | logging |
| 缓存管理 | 内存缓存 |

8. API接口规范

8.1 数据查询接口

* 路径：GET /api/provincial-cpi-data
* 参数：
  + province：省份过滤
  + start\_date/end\_date：时间范围
* 响应：

json

{

"status": "success",

"count": 172,

"data": [{"province":"北京市","date":"2024-05","cpi":103.21}]

}

8.2 数据分析接口

* 路径：GET /api/provincial-cpi-analysis
* 参数：
  + province：指定省份(可选)
* 响应：

json

{

"analysis\_img": "base64...",

"wordcloud\_img": "base64...",

"statistics": {"avg\_cpi":102.34},

"raw\_data": [{"province":"北京市","date":"2024-05","cpi":103.21}]

}

8.3 省份列表接口

* 路径：GET /api/provinces
* 参数：无
* 响应：

json

{

"status": "success",

"provinces": ["北京市", "天津市", ...],

"regions": ["华北", "东北", ...]

}

8.4 健康检查

* 路径：GET /health
* 响应：

json

{

"status": "healthy",

"version": "2.1",

"data\_points": 5428,

"provinces": 31

}

10. 字体处理方案

系统自动处理中文字体显示问题：

1. 预置常见中文字体路径(Windows/Linux/macOS)
2. 启动时检测可用字体
3. 词云和图表使用检测到的字体
4. 无中文字体时使用默认字体并记录告警

本系统为省级CPI数据分析提供完整解决方案，支持时空趋势分析、区域对比和通胀监测，可作为区域经济研究的有效工具。