

相关数据表现：



数据说明：

品牌共现频次：845次

情感相关系数：100%正向关联

网络密度：94.44%（超高连接度）

时段波动：[-30.6%，+92.3%]

## 品牌共生关系

共现矩阵显示：

- 小米→特斯拉单边共现845次（占小米总提及量的**19.3%**）
- 特斯拉未反向提及小米（竞争格局特征）

## 情感共振现象

情感相关系数 = 1.0

- 舆论场呈现"品牌情绪捆绑"效应
- 典型案例：当小米发生刹车事件时，小米汽车讨论区同步出现质量疑虑

## 传播网络特性

网络密度94.44% → 形成紧密的「科技双极」舆论场

## 舆情波动解析

- 峰值期：小米SU7上市引发「特斯拉平替」讨论 【92.3%】
- 低谷期：马斯克访华转移舆论焦点 【-30.6%】

# 代码思路

## 数据预处理篇

### 1. 数据预处理与清洗

- **符号清洗**：通过正则表达式去除微博话题标签（#...#）、特殊方括号内容（【...】）、@用户标记、英文字母和数字等干扰符号
- **表情处理**：专门处理包含在方括号中的表情符号（如[哈哈]）和换行符
- **机械压缩**：使用 `yasuo()` 函数消除重复字符（如将"哈哈哈哈哈"压缩为"哈哈"）
- **空值处理**：删除处理过程中产生的空文本行

### 2. 中文分词处理

- **分词与词性标注**：使用jieba的`pseg.cut`进行词性标注分词
- 词汇过滤
  - 去除停用词（从`stopwords_cn.txt`加载）
  - 保留长度 $\geq 2$ 的纯中文字符
  - 原设计包含词性过滤（名词/形容词/动词），但当前代码中该功能被注释未启用

### 3. 情感分析

- 使用SnowNLP库计算文本情感值（0-1之间的浮点数）
- 三分类规则：
  - $\leq 0.45$  → 负面
  - $0.45-0.55$  → 中性
  - $> 0.55$  → 正面

### 4. 关键技术点

- 正则表达式实现多模式匹配清洗
- 基于词性和停用词表的二次过滤
- 机械压缩算法处理重复字符
- SnowNLP的预训练情感分析模型应用

# 数据分析篇

## 一、数据预处理与分类

### 1. 品牌映射处理

- 通过 `data_process` 函数将分词结果分类为"小米"、"特斯拉"、"品牌共现"三类
- 使用关键词匹配规则：

```
Python"雷总","汽车"等词 → 小米
"自动驾驶","马斯克"等词 → 特斯拉
同时包含"雷/小米"和"斯克/特斯拉" → 品牌共现
```

## 二、品牌关联分析

### 1. 共现矩阵构建：

- 创建2x2矩阵记录品牌共现次数
- 遍历每条评论：

```
Pythonfor 文本 in 数据集:
    识别提及的品牌索引i,j
    if i < j: # 避免重复计数
        矩阵[i][j] += 1
```

- 输出共现矩阵并保存为CSV

### 2. 共现率计算：

计算公式：共现率(A, B) =  $P(A, B) / (P(A) + P(B) - P(A, B))$

$P(A, B)$ ：A和B同时出现的评论数 / 总评论数

$P(A)$ ：A出现的评论数 / 总评论数

$P(B)$ ：B出现的评论数 / 总评论数

## 三、情感关联分析

### 1. 三维情感矩阵

- 创建3D矩阵记录不同品牌组合的情感分布
- 维度说明：
- 使用皮尔逊相关系数：

Python将共现矩阵(2x2)和情感矩阵(2x2x3)展平为一维数组  
计算两者的皮尔逊相关性

## 四、特性关联分析

### 1. 品牌-特性矩阵：

- 构建二维矩阵记录品牌与特性的共现次数
- 计算网络密度：

网络密度 = 实际边数 / 最大可能边数

### 2. 时间序列分析：

- 将数据等分为3个时段
- 计算相邻时段间的共现变化率：
- 计算相邻时间段的共现强度变化率： $(t2 - t1) / t1$ ，其中t1和t2分别是两个时间段的共现强度。  
分析共现强度随时间的变化趋势，揭示用户认知转变。

变化率 = (当期值 - 上期值) / 上期值

## 五、可视化模块

### 1. 网络语义图：

- 多线程处理文本生成词汇共现数据
- 使用networkx构建语义网络
- 采用spring\_layout算法布局节点