统一文本编码格式 确保所有文本数据都采用统一的编码格式（推荐UTF-8），这有助于避免在处理文本时出现编码错误。可以在读取数据时指定编码格式，确保所有文本都是UTF-8编码，核心代码如下。

```
import pandas as pd
df = pd.read_csv('jijihong.csv', encoding='utf-8')
```

```
print(df['text'])
df['text'] = df['text'].str.replace(r'<[^>]+>', '', regex=True)  # 去除HTML标签
df['text'] = df['text'].str.replace(r'http\S+', '', regex=True)  # 去除URLs
df['text'] = df['text'].str.replace(r'\d+', '', regex=True)  # 去除数字
```

```
0        五点一十多去的店里，都没开什么灯，很暗，真的很影响用餐体验，跟我在南昌吃的感觉完全不同。我用...
1        朋友带来吃的，说是霸占了南昌的平价火锅[悠闲] 味道确实不赖，四个女生加了三次菜，其中是三份...
2                                    好吃，服务好经常带小孩一家人去吃，
3        来南昌几天，经常在大街上看到季季红的广告牌，感觉应该是本地比较有名的火锅??，就来尝尝啦 [...
4          季季红火锅，食材新鲜，味道正宗，锅底浓郁，调味恰到好处，服务周到热情，是火锅爱好者的不错选择。
                              ...
10028    第一次去吃结果就踩雷了  猪脑花是冰冻的  牛肉丸脆皮肠也是冰冻的好歹你也冲一下水在给我上呀...
10029    上菜很慢很慢。吃到最后还有菜没上齐 而且分量比其他门店小 真的很无语吃过这么久上菜最慢最慢的...
10030    说句实话 现在服务越来越差 点了个虾滑 我不会下 以前吃喊服务员都会帮忙下 今天我过去吃 吃...
10031                  口味一般 环境一般 服务态度很差  在南昌吃季季红体验感最差的一次了
10032    跨年去他们家吃火锅楼下只有方桌了，我们就两个人表示不想坐方桌，第一个服务员告知我们说楼上没有...
Name: text, Length: 10033, dtype: object
```

```
df['text'] = df['text'].str.replace(r'\b\d{11}\b', '', regex=True)
```

```
import jieba
jieba.add_word("季季红",freq=100)
df['text'] = df['text'].apply(lambda x: ' '.join(jieba.cut(x)))
```

```
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\Administrator\AppData\Local\Temp\jieba.cache
Loading model cost 0.581 seconds.
Prefix dict has been built successfully.
```

```
df['text'].to_csv("data_jieba.csv",index=False,header=True,encoding='utf-8-sig')
```

```
stopwords = set(open('stopwords_hit.txt', 'r', encoding='utf-8').read().split())
df['text'] = df['text'].apply(lambda x: ' '.join(word for word in x.split() if word not in stopwords))
df['text']
```

```
0        五点 十多 去 店里 都 没开 灯 很 暗 真的 很 影响 用餐 体验 南昌 吃 感觉 完全...
1        朋友 带来 吃 说 霸占 南昌 平价 火锅 悠闲 味道 确实 不赖 四个 女生 加 三次 菜...
2                                好吃 服务 好 经常 带 小孩 一家人 去 吃
3        南昌 几天 经常 大街 上 看到 季季红 广告牌 感觉 应该 本地 比较 有名 火锅 尝尝 ...
4        季季红 火锅 食材 新鲜 味道 正宗 锅底 浓郁 调味 恰到好处 服务周到 热情 火锅 爱好...
                              ...
10028    第一次 去 吃 踩 雷 猪脑 花是 冰冻 牛肉丸 脆皮 肠 冰冻 好歹 一下 水在 上 冰箱...
10029    上菜 很慢 很慢 吃 最后 菜 没 上 齐 分量 门店 小 真的 很无语 吃 这么久 上菜 ...
10030    说句实话 现在 服务 越来越 差 点 虾 滑 不会 下 以前 吃 喊 服务员 都 会 帮忙 ...
10031                  口味 环境 服务态度 很差 南昌 吃 季季红 体验 感 最差 一次
10032    跨年 去 家 吃火锅 楼下 方桌 两个 人 表示 不想 坐 方桌 第一个 服务员 告知 说 ...
Name: text, Length: 10033, dtype: object
```

```
# 构建同义词典
synonym_dict = {
        '好吃': '美味',
        '赞': '美味',
        '绝味': '美味',
        '满意': '喜欢',
        '合口': '美味',    # 假设'合口'是方言表达，可以替换为'美味'
        # ... 其他同义词映射
}

# 应用同义词典，将文本中的同义词替换为标准表达
df['text'] = df['text'].apply(lambda x: ' '.join(synonym_dict.get(word, word) for word in x.split()))

# 处理完成后的df['text']将包含统一的标准表达
df['text'].to_csv('data_syn.csv',index=False,header=True,encoding='utf-8-sig')
df['text']
```

```
0        五点 十多 去 店里 都 没开 灯 很 暗 真的 很 影响 用餐 体验 南昌 吃 感觉 完全...
1        朋友 带来 吃 说 霸占 南昌 平价 火锅 悠闲 味道 确实 不赖 四个 女生 加 三次 菜...
2                                美味 服务 好 经常 带 小孩 一家人 去 吃
3        南昌 几天 经常 大街 上 看到 季季红 广告牌 感觉 应该 本地 比较 有名 火锅 尝尝 ...
4        季季红 火锅 食材 新鲜 味道 正宗 锅底 浓郁 调味 恰到好处 服务周到 热情 火锅 爱好...
                              ...
10028    第一次 去 吃 踩 雷 猪脑 花是 冰冻 牛肉丸 脆皮 肠 冰冻 好歹 一下 水在 上 冰箱...
10029    上菜 很慢 很慢 吃 最后 菜 没 上 齐 分量 门店 小 真的 很无语 吃 这么久 上菜 ...
```

```
10030    说句实话 现在 服务 越来越 差 点 虾 滑 不会 下 以前 吃 喊 服务员 都 会 帮忙 ...
10031                    口味 环境 服务态度 很差 南昌 吃 季季红 体验 感 最差 一次
10032    跨年 去 家 吃火锅 楼下 方桌 两个 人 表示 不想 坐 方桌 第一个 服务员 告知 说 ...
Name: text, Length: 10033, dtype: object
```

```
from snownlp import SnowNLP
df = pd.read_csv("data_syn.csv",encoding='utf')
# 先将所有非字符串类型的文本转换为字符串，然后去除或替换NaN值
df['text'] = df['text'].astype(str)
# 去除包含NaN文本的行
df.dropna(subset=['text'], inplace=True)
# SnowNLP的情感分析返回一个介于0到1之间的分数，大于0.5通常表示正面情绪
df['sentiment_score'] = df['text'].apply(lambda x: SnowNLP(x).sentiments)
# 将情感分数转换为情感标签
df['sentiment_label'] = df['sentiment_score'].apply(lambda x: '正面' if x > 0.5 else '负面')
```

```
df['sentiment_label']
# 统计正面和负面情感的数量
sentiment_counts = df['sentiment_label'].value_counts()

# 打印情感标签的计数结果
print(sentiment_counts)

# 计算正负情感的百分比
total_reviews = len(df)
positive_percent = (sentiment_counts.get('正面', 0) / total_reviews) * 100
negative_percent = (sentiment_counts.get('负面', 0) / total_reviews) * 100

# 打印情感百分比
print(f"正面情感占比：{positive_percent:.2f}%")
print(f"负面情感占比：{negative_percent:.2f}%")
```

```
sentiment_label
正面    8503
负面    1530
Name: count, dtype: int64
正面情感占比：84.75%
负面情感占比：15.25%
```

```
#将 df['sentiment_score'] 和 df['sentiment_label'] 这两列数据保存到csv文件
df[['sentiment_score','sentiment_label']].to_csv('sentiment_data.csv', index=False, encoding='utf-8-sig')
```