import jieba

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.feature\_extraction.text import TfidfTransformer

from sklearn.cluster import Birch

import warnings

warnings.filterwarnings("ignore")

# 读取文本数据

with open('test1.txt', 'r', encoding='utf-8') as f:

documents = f.readlines()[:100]

# 读取停用词

stopword\_path = "stop\_words.txt"

with open(stopword\_path, 'r', encoding='utf-8') as f:

stop\_words = [line.strip() for line in f]

# 分词处理

corpus = []

for doc in documents:

words = jieba.cut(doc.strip())

corpus.append(' '.join(words))

# print(corpus)

# 4.1 文本转换成词袋模型(词频作为统计指标) 加载停用词,添加词语进词袋时会过滤停用词

countVectorizer = CountVectorizer(stop\_words=stop\_words, analyzer="word")

count\_v = countVectorizer.fit\_transform(corpus)

# 4.2 词频统计指标转换 tf-idf统计指标 (不是必须的,用哪种指标根据具体业务来看)

tfidfTransformer = TfidfTransformer()

tfidf = tfidfTransformer.fit\_transform(count\_v)

# print(tfidf.toarray())

# birch聚类算法

'''

tf-idf统计指标作为数据

将参数设为None, 自动分类

'''

birch\_cluster = Birch(n\_clusters=None)

birch\_result = birch\_cluster.fit\_predict(tfidf)

print("Predicting result: ", birch\_result)

print(f'num\_clusters: {max(birch\_result)+1}')

# 自动划分为92类