1. 简述文本数据的特征提取、汉语分词和词包模型技术摘要（不要超过**300字**，逻辑要通，**要有小标题**，要排版清晰，请不要网络大幅摘抄）

**一、文本数据的特征提取**文本数据的特征提取是将文本转换为可用于机器学习模型的数值型特征的过程。常用方法包括词包模型、TF-IDF 等。这些方法可以将文本转换为向量表示，以便计算机能够理解和处理。

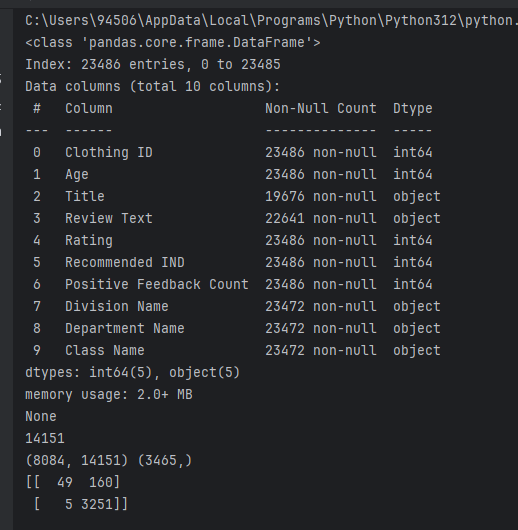
**二、汉语分词**汉语分词是将汉字序列切分成有意义的词语的过程。对于中文文本，分词是文本处理的基础步骤。常用的分词工具包括 jieba、THULAC 等，它们能够根据语料库和规则将汉字序列切分成词语。**三、词包模型**

词包模型是一种简单而有效的文本表示方法。它将文本看作是无序的词语集合，忽略了词语之间的顺序和语法。每个文本可以表示为一个向量，向量的每个维度对应一个词语，在文本中出现则对应维度的值为1，否则为0。词包模型忽略了词语的顺序，适用于很多文本分类和聚类任务。

**二、利用算法解决问题**

1.将教材中实战项目“女装电子商务数据集”进行完善，解决数据不平衡的问题。

import pandas as pd  
from sklearn.feature\_extraction.text import CountVectorizer  
from sklearn.model\_selection import train\_test\_split  
import warnings  
warnings.filterwarnings("ignore")  
  
# 导入数据集  
raw\_data = pd.read\_csv('Womens Clothing E-Commerce Reviews.csv', index\_col = 0)  
print(raw\_data.info())  
  
import string  
  
# 文本预处理函数：去除标点符号  
def text\_process(text\_data):  
 nopunc = [word for word in text\_data if word not in string.punctuation]  
 nopunc = ''.join(nopunc)  
 return [word for word in nopunc.split()]  
  
# 对 'Review Text' 列的前5个样本应用文本预处理函数  
raw\_data['Review Text'].head().apply(text\_process)  
  
# 删除缺失值  
df = raw\_data.dropna(axis = 0)  
  
# 仅保留评分为 1 或 5 的样本  
df\_binomial = df[(df['Rating'] == 5) | (df['Rating'] == 1)]  
x = df\_binomial['Review Text']  
y = df\_binomial['Rating']  
  
# 创建文本向量化器  
transformer = CountVectorizer(analyzer = text\_process, ngram\_range=(2,2), stop\_words = 'english').fit(x)  
print(len(transformer.vocabulary\_))  
  
# 将文本转换为向量  
x = transformer.transform(x)  
  
# 划分训练集和测试集  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.3, random\_state = 42)  
print(x\_train.shape, y\_test.shape)  
  
from sklearn.naive\_bayes import MultinomialNB  
  
# 使用多项式朴素贝叶斯分类器  
clf = MultinomialNB()  
clf.fit(x\_train, y\_train)  
  
from sklearn.metrics import confusion\_matrix, classification\_report  
  
# 在测试集上进行预测并打印混淆矩阵和分类报告  
predict = clf.predict(x\_test)  
print(confusion\_matrix(y\_test, predict))  
print('\n')  
print(classification\_report(y\_test, predict))  
  
from imblearn.over\_sampling import SMOTE  
  
# 使用 SMOTE 过采样算法处理训练集  
smote = SMOTE(random\_state=42)  
x\_train\_resampled, y\_train\_resampled = smote.fit\_resample(x\_train, y\_train)  
  
# 使用新的平衡数据集重新训练模型  
clf.fit(x\_train\_resampled, y\_train\_resampled)  
  
# 使用新模型预测测试集  
predict = clf.predict(x\_test)  
  
# 打印混淆矩阵和分类报告  
print(confusion\_matrix(y\_test, predict))  
print('\n')  
print(classification\_report(y\_test, predict))



日历

描述已自动生成使用SMOTE算法对训练集进行过采样，以解决评分不平衡的问题，重新使用过采样后的数据训练模型，并在测试集上进行评估。

通过比较两次结果，可以评估SMOTE有效地改善了模型的性能，尤其是在识别少数类别的评论（评分为1）方面。

2.自行解决一个中文文本数据分析问题。

**解决中文外卖评论数据集**

import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.feature\_extraction.text import CountVectorizer, TfidfTransformer  
from sklearn.naive\_bayes import MultinomialNB  
from sklearn.pipeline import Pipeline  
from sklearn.metrics import classification\_report  
import jieba  
  
# 数据预处理的函数  
def preprocess\_text(text):  
 # 使用jieba进行分词处理  
 words = jieba.cut(text)  
 return ' '.join(words)  
  
# 读取数据集  
df = pd.read\_csv("中文外卖评论数据集.csv")  
df['review\_clean'] = df['review'].apply(preprocess\_text)  
  
# 划分数据集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['review\_clean'], df['label'], test\_size=0.2, random\_state=42)  
  
# 创建管道，包括向量化和分类器  
text\_clf = Pipeline([  
 ('vect', CountVectorizer()),  
 ('tfidf', TfidfTransformer()),  
 ('clf', MultinomialNB()),  
])  
  
# 训练模型  
text\_clf.fit(X\_train, y\_train)  
# 预测  
predicted = text\_clf.predict(X\_test)  
# 输出评估结果  
print(classification\_report(y\_test, predicted))  
  
# 模拟输入的中文评论  
input\_review = '很好吃！'  
  
# 预处理函数  
def preprocess\_text(text):  
 words = jieba.cut(text)  
 return ' '.join(words)  
  
# 对评论进行预处理和特征提取  
processed\_review = preprocess\_text(input\_review)  
  
# 使用模型进行预测  
prediction = text\_clf.predict([processed\_review])  
  
# 一般情况下，标签是在训练模型时定义的，通常1代表正面评论，0代表负面评论  
print('输入的评价为：',input\_review)  
if prediction[0] == 1:  
 print("该评价为好评")  
else:  
 print("该评价为差评")

**结果：**

日历

描述已自动生成

对于评价结果，代码输出了分类报告，其中包括了准确率、召回率、F1-score等指标，用于评估模型在每个类别上的性能表现。同时，对于新输入的评论，根据模型的预测结果，输出了评论的情感极性（好评或差评）。

要求：

1）针对当前数据、模型，**详叙数据预处理、模型参数调整过程及效果比对**；

2）代码要列出，重点代码加注释说明，特别是自己调试过程中的自我理解；

3）运行结果要截图，结果要文字说明；

4）注意排版