Python与金融数据挖掘(5)

文欣秀

wenxinxiu@ecust.edu.cn

英文文本分析

AI in Finance: Challenges, Techniques, and Opportunities

LONGBING CAO, University of Technology Sydney, Australia

AI in finance refers to the applications of AI techniques in financial businesses. This area has attracted attention for decades, with both classic and modern AI techniques applied to increasingly broader areas of finance, economy, and society. In contrast to reviews on discussing the problems, aspects, and opportunities of finance benefited from specific or some new-generation AI and data science (AIDS) techniques or the progress of applying specific techniques to resolving certain financial problems, this review offers a comprehensive and dense landscape of the overwhelming challenges, techniques, and opportunities of AIDS research in finance over the past decades. The challenges of financial businesses and data are first outlined, followed by a comprehensive categorization and a dense overview of the decades of AIDS research in finance. We then structure and illustrate the data-driven analytics and learning of financial businesses and data. A comparison, criticism, and discussion of classic versus modern AIDS techniques for finance follows. Finally, the open issues and opportunities to address future AIDS-empowered finance and finance-motivated AIDS research are discussed.

代码实现（一）

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from imageio.v2 import imread
import string
fobj = open("AI_in_Finance.txt", "r")
paper=fobj. read()
paper = paper. lower()
for ch in string.punctuation:
    paper = paper. replace(ch, " ") #将特殊字符替换为空格
words = paper.split( )
```

代码实现（二）

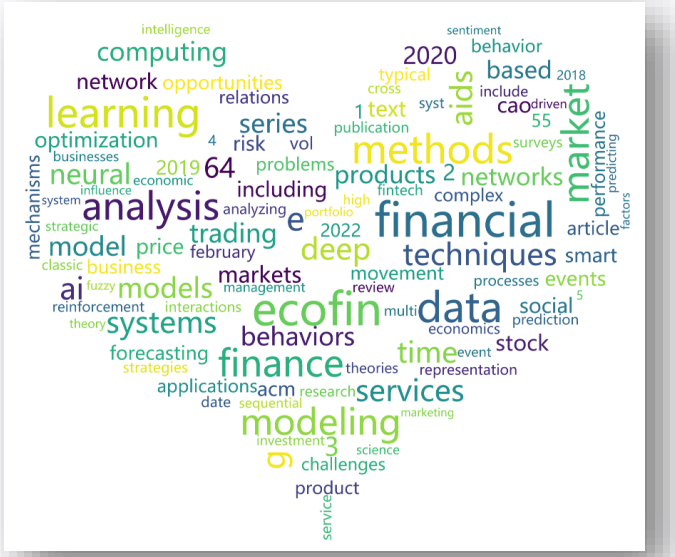
```
excludes=set()
with open("stop.txt", "r") as handle:
    for i in handle:
      i=i. strip()
      excludes. add(i)
counts = { }
for word in words:
    if word in excludes:
        continue
    else:
        counts[word] = counts. get(word,0) + 1
```

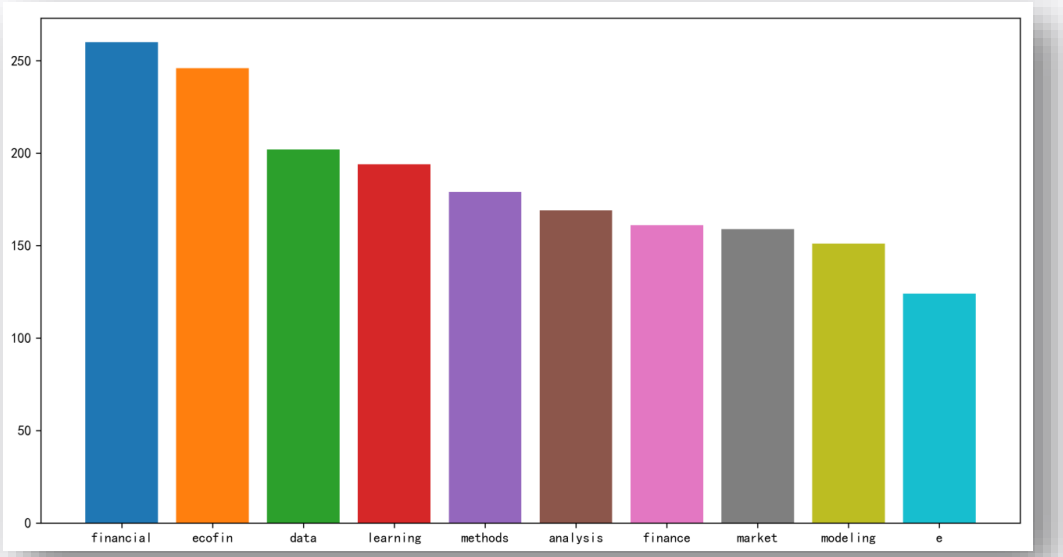
代码实现（三）

```
pic = imread('cloud.jpg')
wc=WordCloud(mask=pic,font_path='msyh.ttc',  #中文字体
        repeat=False, background_color='white', #设置背景颜色
        max_words=110, max_font_size=120,    #设置字体最大值
        min_font_size=10, random_state=50,     #设置配色方案
        scale=10)
wc.generate_from_frequencies(counts)
plt.imshow(wc) #将数值以图片形式显示出来
plt.show()
```

模块化程序设计

将一个大型程序按照其功能分解成若干个相对独立的功能模块分别进行设计，最后把这些功能模块按层次关系进行组装。





函 数 定 义

```
def  函数名 ([形式参数列表]):
        执行语句
        [return 返回值]
```

参 数 传 递

- 位置传递
- 关键字传递
- 默认值参数传递
- 元组传递
- 字典传递

lambda匿名函数

lambda： Python预留的关键字，可以定义一个匿名函数，函数体是一个简单的表达式而不是语句块，通常用在只使用一次的场景

形　式： lambda argument_list: expression

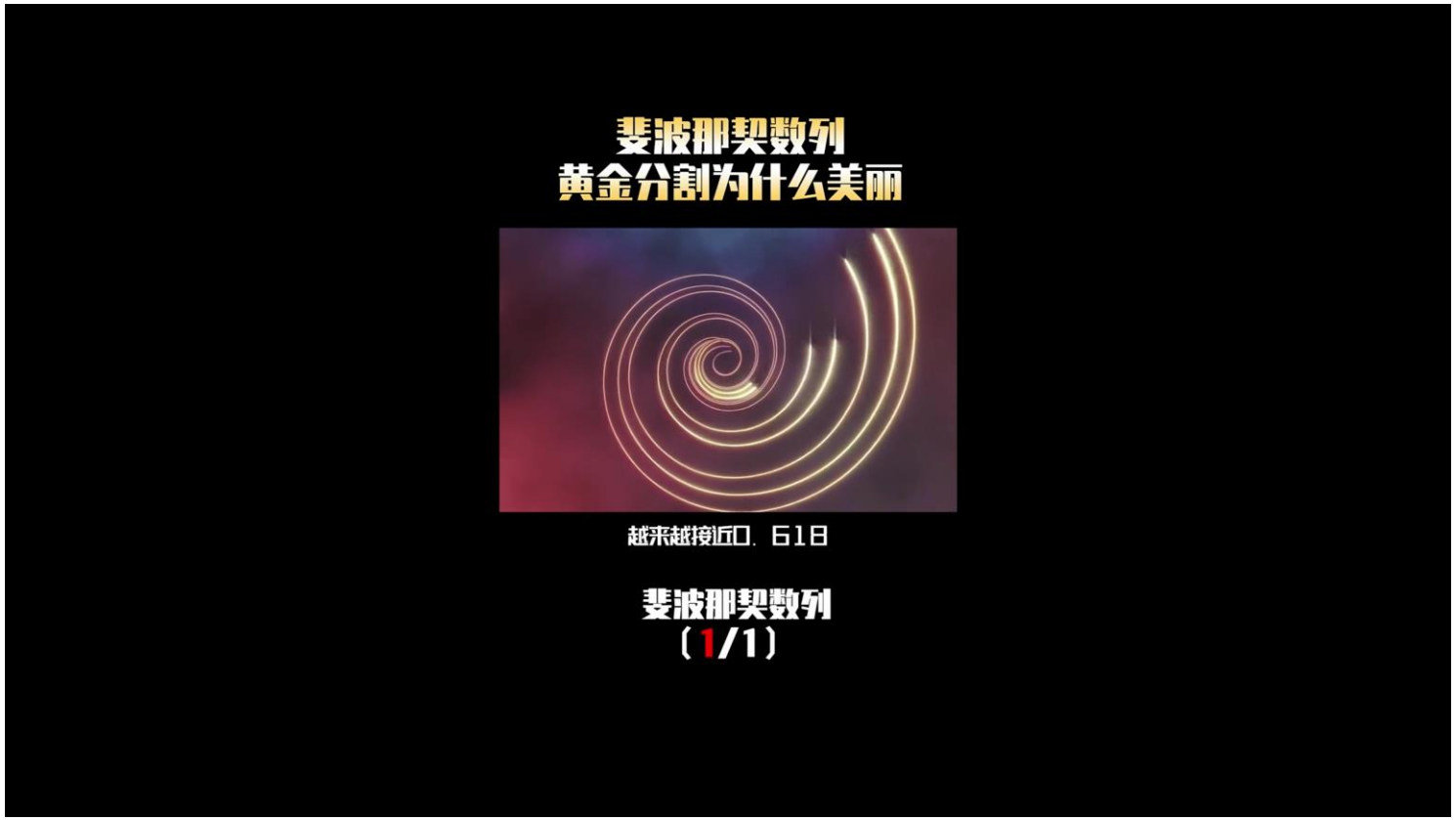
例　子： **lambda x: x**3** 输入x，输出x的3次方

map( ) 函数

- 接收一个函数 f 和一个或多个list，并通过把函数 f 依次作用在list 的每个元素上，得到一个新的 list 并返回。

```
>>> price=[35.8, 24.9, 23.5]
>>> list(map(lambda x: x*2, price))
>>> price1=[35.8, 24.9, 23.5]
>>> price2=[88.5, 32.0, 12.9]
>>> list(map(lambda x,y: x+y, price1, price2))
```

案例分析



函数的递归调用

- 函数在自身的定义中又调用自身
- 分析递归问题的关键

递推公式： 每次调用自身时参数的变化规律

结束条件： 递归调用结束的条件

斐波那契数列

```
def fibs(num):
    result=[1,1]
    for i in range(num-2):
        result.append(result[-2]+result[-1])
    return result
print(fibs(20))
```

```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 23
3, 377, 610, 987, 1597, 2584, 4181, 6765]
```

斐波那契数列

```
def fibs(num):
    if num==1:
        return 1
    elif num==2:
        return 1
    else:
        return fibs(num-1)+fibs(num-2)
for i in range(1,21):
    print(fibs(i),end=" ")
```

```
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
1597 2584 4181 6765
```

变量的作用域

变量作用域：指变量能被有效使用的范围

全局变量：在函数之外定义的变量，在整个程序范围内起作用

局部变量：指在某个函数内部定义的变量，在该函数范围内起作用

变量的作用域

说明： 全局变量与局部变量同名时，函数内部局部变量起作用，函数外部全局变量起作用

```
a, b=5, 6
def fun():
    a, b=10, 15
    print("函数内： a=%d, b=%d" % (a, b))
fun()
print("函数外： a=%d, b=%d" % (a, b))
```

全局变量

```
C=0  #在文件开头声明全局变量
def modifyConstant() :
    global C
    print(C)
    C+=1
    return

if __name__ == '__main__' :
    modifyConstant()
    print(C)
```

全局变量

```
#b.py 把全局变量定义在一个单独的模块中
gl_1 = 'hello'
gl_2 = 'world'

#a.py 在其它模块中使用
import b
def hello_world():
    print(b.gl_1,b.gl_2)

if __name__ == '__main__' :
    hello_world()
```

课堂练习

以下程序的执行结果是（ ）

```
result = 2
def test(result, m):
    result= pow(result, m)
    print(result, end=" ")
test(result,3)
print(result)
```

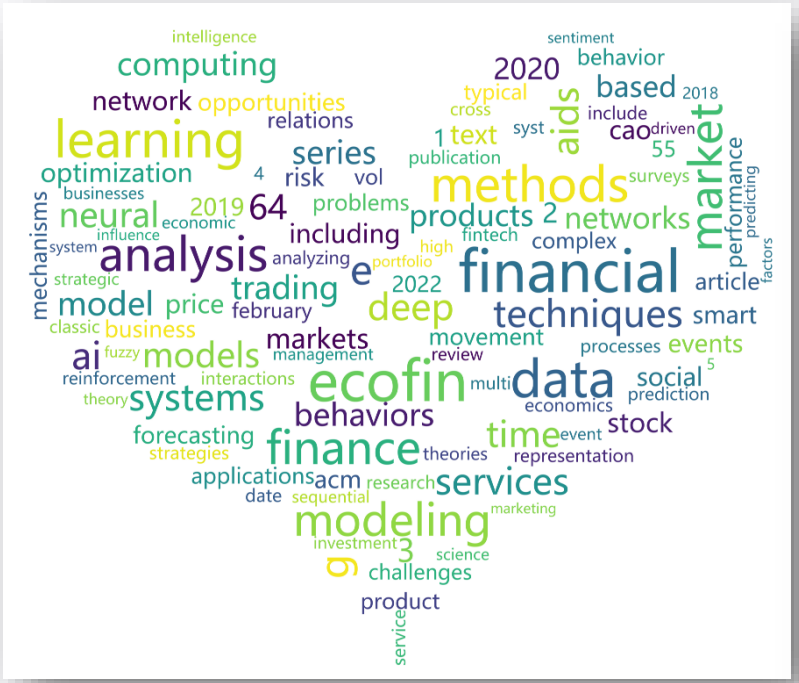
A、8 2

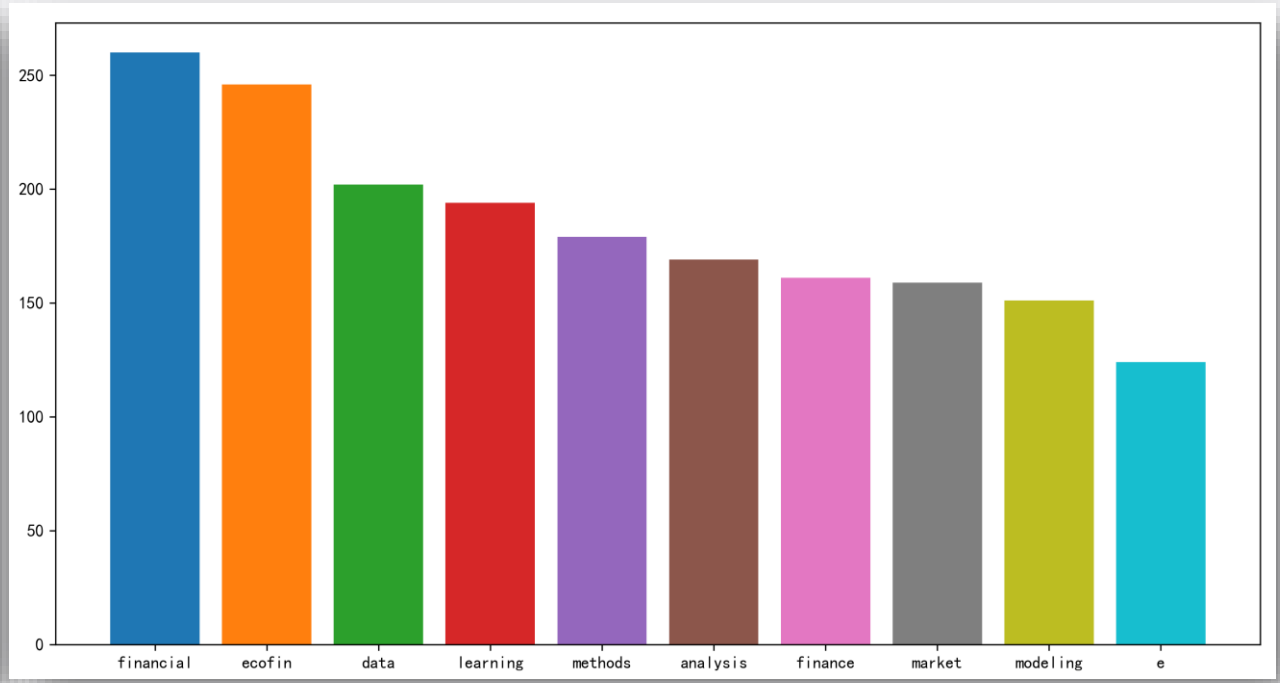
B、2 2

C、2 8

D、8 8

案例分析


intelligence
computing
sentiment
2020
behavior
network
opportunities
typical
based
2018
cross
include
relations
aids
learning
1
text
syst
cao
driven
market
series
publication
55
optimization
4
risk
vol
methods
surveys
businesses
2019
64
problems
products
2
networks
performance
predicting
mechanisms
neural
economic
including
fintech
complex
influence
high
system
analysis
analyzing
portfolio
e
financial
article
factors
strategic
trading
2022
deep
techniques
smart
model
price
february
classic
business
movement
processes
events
markets
review
ai
fuzzy
models
management
5
reinforcement
interactions
ecofin
multi
data
social
prediction
theory
systems
economics
behaviors
stock
forecasting
time
event
finance
theories
representation
strategies
applications
acm
research
services
date
sequential
marketing
modeling
investment
3
science
challenges
product
service


250
200
150
100
50
0
financial
ecofin
data
learning
methods
analysis
finance
market
modeling
e

代码实现（一）

```
import matplotlib.pyplot as plt

from wordcloud import WordCloud

from imageio.v2 import imread

import string
```

代码实现（二）

```
def stat(filename):
    fobj = open(filename, "r")
    paper=fobj.read()
    paper = paper.lower()
    for ch in string.punctuation:
        paper = paper.replace(ch, " ") #将特殊字符替换为空格
    words = paper.split( )
    excludes=set()
    with open("stop.txt","r") as handle:
        for i in handle:
          i=i. strip()
          excludes. add(i)
```

代码实现（二）

```
def stat(filename):
    #接上页…
    counts = { }
    for word in words:
        if word in excludes:
            continue
        counts[word] = counts. get(word,0) + 1
    return counts
```

代码实现（三）

```
def bar(counts):
    items=list(counts.items())
    items.sort(key=lambda x:x[1],reverse=True)
    plt.rcParams['font.sans-serif']=['SimHei']   #用来正常显示中文标签
    for i in range(10):
        word,count=items[i]
        print("{0:<10}{1:>5}".format(word,count))
        plt.bar(word,count)
    plt.show()
```

代码实现（四）

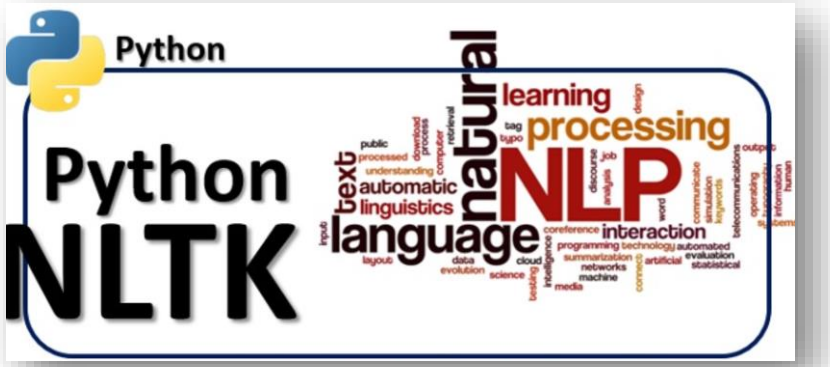
```
def cloud(counts, mask):
    pic = imread(mask)
    wc=WordCloud(mask=pic,font_path='msyh.ttc',  #中文字体
                 repeat=False, background_color='white', #设置背景颜色
                 max_words=110, max_font_size=120,    #设置字体最大值
                 min_font_size=10, random_state=50,     #设置配色方案
                 scale=10)
    wc.generate_from_frequencies(counts)
    plt.imshow(wc) #将数值以图片形式显示出来
    plt.show()
```

代码实现（六）

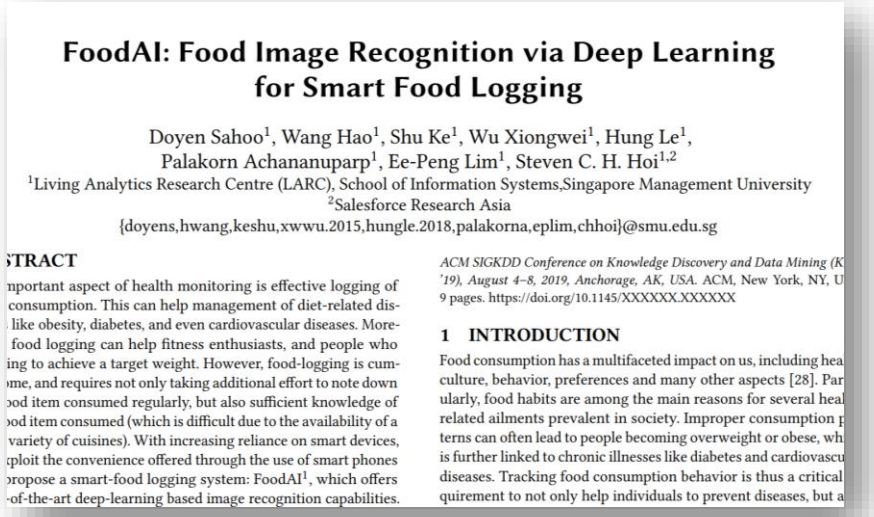
```
name=input("请输入文献名：")
counts=stat(name)
choice=input("请输入您的选择：")
if choice=="词云":
    mask=input("请输入图片名:")
    cloud(counts, mask)
else:
    bar(counts)
```

NLTK

NLTK（Natural Language Toolkit）是一个强大的Python库，专门为处理人类语言数据而设计。它提供了一系列丰富的资源和工具，包括文本处理、语法分析、语义推理和机器学习等。



NLTK



FoodAI: Food Image Recognition via Deep Learning for Smart Food Logging

Doyen Sahoo[1], Wang Hao[1], Shu Ke[1], Wu Xiongwei[1], Hung Le[1],
Palakorn Achananuparp[1], Ee-Peng Lim[1], Steven C. H. Hoi[1,2]
[1]Living Analytics Research Centre (LARC), School of Information Systems,Singapore Management University
[2]Salesforce Research Asia
{doyens,hwang,keshu,xwwu.2015,hungle.2018,palakorna,eplim,chhoi}@smu.edu.sg

STRACT

mportant aspect of health monitoring is effective logging of consumption. This can help management of diet-related dis- like obesity, diabetes, and even cardiovascular diseases. More- food logging can help fitness enthusiasts, and people who ing to achieve a target weight. However, food-logging is cum- me, and requires not only taking additional effort to note down od item consumed regularly, but also sufficient knowledge of od item consumed (which is difficult due to the availability of a variety of cuisines). With increasing reliance on smart devices, xploit the convenience offered through the use of smart phones propose a smart-food logging system: FoodAI[1], which offers -of-the-art deep-learning based image recognition capabilities.

ACM SIGKDD Conference on Knowledge Discovery and Data Mining (K '19), August 4–8, 2019, Anchorage, AK, USA. ACM, New York, NY, U 9 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Food consumption has a multifaceted impact on us, including hea culture, behavior, preferences and many other aspects [28]. Par ularly, food habits are among the main reasons for several hea related ailments prevalent in society. Improper consumption p terns can often lead to people becoming overweight or obese, wh is further linked to chronic illnesses like diabetes and cardiovascu diseases. Tracking food consumption behavior is thus a critical quirement to not only help individuals to prevent diseases, but a

思考： 如何用NLTK库读取文件并进行词频统计？

NLTK库的数据下载

- import nltk
- nltk. download('punkt') #用于句子分割和单词分割
- nltk. download('vader_lexicon') #情感词典
- nltk. download('stopwords') #停用词资源
- nltk. download('averaged_perceptron_tagger') #词性标注器
- nltk.download('treebank') #treebank大型语料库
- …

NLTK分词

```
from nltk.tokenize import word_tokenize

text = "It is an example sentence."

tokens = word_tokenize(text) #实现分词操作

print(tokens)
```

NLTK标点符号过滤

```
from nltk.tokenize import word_tokenize
import string
text = input("sentence:")
tokens = word_tokenize(text)
print(tokens)
#过滤标点符号
filtered_tokens = [w for w in tokens if w not in string.punctuation]
print(filtered_tokens)
```

NLTK停用词过滤

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
text = input("sentence:")
tokens = word_tokenize(text)
print(tokens)
stop_words = set(stopwords.words('english')) #英文停用词集合
filtered_tokens = [w for w in tokens if w.lower() not in stop_words]
print(filtered_tokens)
```

NLTK频率分布

```
from nltk.tokenize import word_tokenize
from nltk import FreqDist
text = input("sentence:")
tokens = word_tokenize(text)
print(tokens)
fdist = FreqDist(tokens) #生成词频字典
print(fdist.most_common(5))
```

代码实现（一）

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk import FreqDist
import string
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from imageio.v2 import imread
with open("food_AI.txt","r",) as fobj:
    paper=fobj.read()
```

代码实现（二）

```
text = word_tokenize(paper)
result = [w for w in text if w not in string.punctuation]
stop_words = set(stopwords.words('english')) #英文停用词集合
info = [w for w in result if w.lower() not in stop_words]
counts = FreqDist(info) #生成词频字典
pic = imread('brain.png')
```

代码实现（三）

```
pic = imread('brain.png')
wc=WordCloud(mask=pic,font_path='msyh.ttc',  #中文字体
        repeat=False, background_color='white', #设置背景颜色
        max_words=110, max_font_size=120,    #设置字体最大值
        min_font_size=10, random_state=50,     #设置配色方案
        scale=10)
wc.generate_from_frequencies(counts)
plt.imshow(wc) #将数值以图片形式显示出来
plt.show()
```

案例分析


finance.sina.com.cn
新浪首页 >> 蔡崇信反思阿里 海通国际巨亏从何而来 金价再创纪录新高
sina 新浪财经
手机版
新浪财经APP 新浪科技 投教基地
黑猫投诉
领跑者
免费试用
买基
股票 新股 港股 美股 基金 期货 外汇 黄金 债券 理财 银行 保险 信托 医药 科创板 ESG
环球股指 黄金 原油 投研中心 新浪财经APP
行情
简称/代码/拼音
sina 新浪财经
打破信息不对称
7x24快讯
16:33
采纳股份：美国FDA宣布对全资子公司发出…

HTML

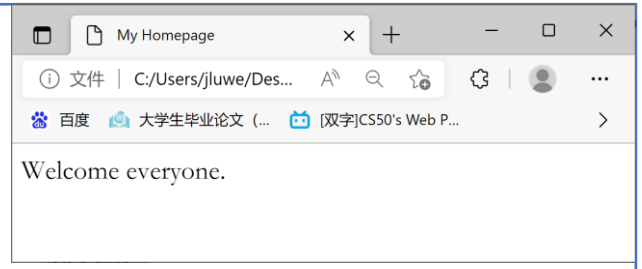
```
<html>
    <head>
        <title>My Homepage</title>
    </head>
    <body>
        Welcome everyone.
    </body>
</html>
```

Head

- <title>
- <meta>
- <link>
- <script>

<meta>标签

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="author" content="Mike">
<title>My Homepage</title>
</head>
<body>
Welcome everyone.
</body>
</html>
```

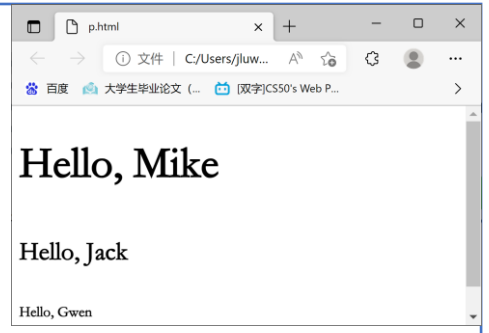


Body

- <p>
-
-
-
- <img>
- <a>

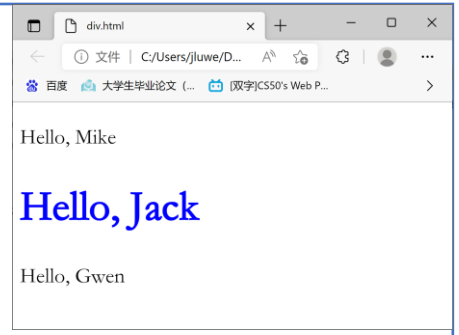
<p>标记

```
<!DOCTYPE html>
<html>
<body>
<p><h1>Hello, Mike</h1></p>
<p><h3>Hello, Jack</h3></p>
<p><h6>Hello, Gwen</h6></p>
</body>
</html>
```



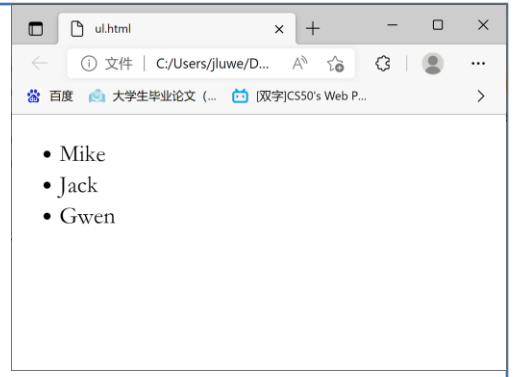
标记

```
<!DOCTYPE html>
<html>
<body>
<p>Hello, Mike</p>
<div style="color:#0000FF">
  <p><h1>Hello, Jack</h1></p>
</div>
<p>Hello, Gwen</p>
</body>
</html>
```



标记

```
<!DOCTYPE html>
<html>
<body>
<ul>
    <li>Mike</li>
    <li>Jack</li>
    <li>Gwen</li>
</ul>
</body>
</html>
```

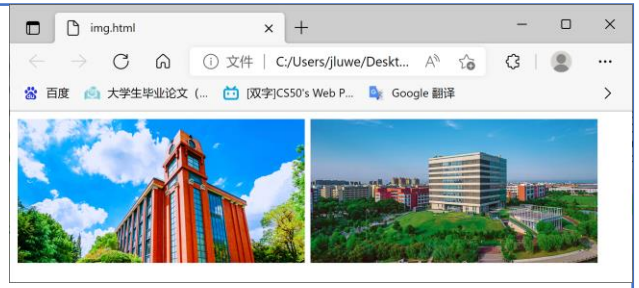


 标记

```
<!DOCTYPE html>
<html>
<body>
<ol>
    <li>Mike</li>
    <li>Jack</li>
    <li>Gwen</li>
</ol>
</body>
</html>
```

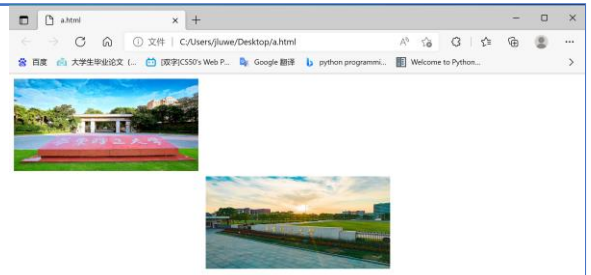
<img>标记

```
<!DOCTYPE html>
<html>
<body>
<img src="XH.jpg" alt="XH" width="300" height="150">
<img src="FX.jpg" alt="FX" width="300" height="150">
</body>
</html>
```



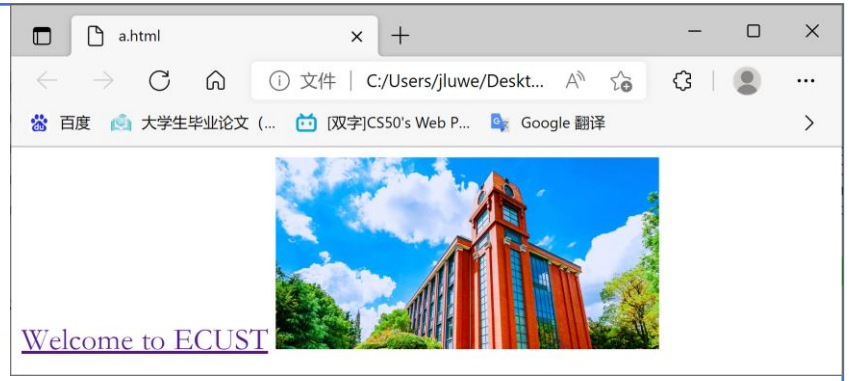
和 <img>标记

```
<!DOCTYPE html>
<html>
<body>
<img src="XH.jpg" alt="XH" width="300" height="150">
<div align="center">
<img src="FX.jpg" alt="FX" width="300" height="150">
</div>
</body>
</html>
```



<a> 标记

```
<!DOCTYPE html>
<html>
<body>
<a href="http://www.ecust.edu.cn">Welcome to ECUST</a>
<a href="http://xiaoban.ecust.edu.cn"><img src="XH.jpg"
                alt="XH" width="300" height="150"> </a>
</body>
</html>
```



网络爬虫

网络爬虫： 是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。

爬虫常用库： requests

urllib

beautifulsoup

…

简单爬虫程序基本步骤

- 获取网页源码
- 根据源码中所在链接特点写出正则表达式
- 用正则对象匹配获取目标链接
- 用循环结构遍历目标链接并自动下载信息

爬取网页内容

requests第三方库：用pip install requests安装

```
import requests
url="https://finance.sina.com.cn/"
r=requests.get(url)
print(r.text)
```

Requests对象的属性

属性	说明
r. text	网页响应内容的**字符串**形式
r. encoding	**猜测**网页响应内容编码方式
r. apparent_encoding	从网页内容**分析**出编码方式
r. content	网页响应内容的**二进制**形式

修改编码方式

```
import requests
url="https://finance.sina.com.cn/"
r=requests.get(url)
r.encoding=r.apparent_encoding
data=r.text
print(data)
```

爬取网页内容并存入文件

```
import requests
url="https://finance.sina.com.cn/"
r=requests.get(url)
r.encoding=r.apparent_encoding
data=r.text
fobj=open("result.txt", "w", encoding="utf-8")
fobj.write(data)
fobj.close()
```

爬取单张图片

```
import requests
url="https://n.sinaimg.cn/finance/transform/562/w360h202/20240407/
b3a6-3326da7985b4ad46379181eb60ac28db.jpg"
r=requests.get(url)
data=r.content
fobj=open("result.jpg","wb")
fobj.write(data)
fobj.close()
```

问题： 如何获取多张图片并保存到硬盘？

爬取单个PDF文件

```
import requests
url="https://staticpacific.blob.core.windows.net/\
press-releases-attachments/3540725/\
HKEX-EPS_20250325_11582990_0.PDF"
r=requests.get(url)
data=r.content
fobj=open("小米发布.pdf","wb")
fobj.write(data)
fobj.close()
```

问题： 如何获取多个pdf文档内容？

正则表达式

正则表达式（regular expression，re）:是由一些**特定字符**及其**组合**所组成的**字符串表达式（模板）**，用来对目标字符串进行过滤操作。处理正则表达式需要引入**标准库是re**。

字符串匹配模式

（1）数字和字符

'**\d**'：匹配一个数字

'**\w**'：匹配一个字母、数字或下划线

例如：'\d\d\d'能匹配'021'，但不能匹配'AB1'

'\w\w\w'既能匹配'021'，又能匹配'AB1'

匹配数字与字符案例

```
import re
content = 'Hello 123 world 456'
result = re. findall('\d\d\d', content)
print(result)
```

```
import re
content = 'Hello 123 world 456'
result = re. findall('\w\w\w', content)
print(result)
```

字符串匹配模式

（2）任意单个字符

'.'：匹配任意单个字符

```
import re
content = 'Hello 123 world 456'
result = re. findall('.o', content)
print(result)
```

字符串匹配模式

（3）多个字符

'*'：表示任意多个字符(包括0个)

'+'：表示至少1个字符

'?'：表示0个或1个字符

'{n}'：表示n个字符

'{n, m}'：表示n至m个字符

匹配数字与字符案例

```
import re
content = 'Hello 123 world 456'
result = re. findall('.{1,8}', content)
print(result)
```

```
import re
content = 'Hello 123 world 456'
result = re. findall('.{5}', content)
print(result)
```

字符串匹配模式

（4）字符范围

'**[ ]**'：表示字符范围，1组方括号只能表示1个字符

例如：'[0-9a-zA-Z_]'匹配1个数字、字母或下划线

'|'：表示或关系

例如：'[P|p]ython'可匹配'Python'或'python'

案例分析

```
import re
content = 'Hello 123  world 456'
result = re. findall('[0-9a-zA-Z\_]{5}', content)
print(result)
```

字符串匹配模式

（5）开头和结尾

'^'：表示行开头，'^\d'表示必须以数字开头

'$'：表示行结束，'\d$'表示必须以数字结束

例如：'^py$'只能匹配整行完整的'py'

字符串匹配模式

（6）特殊字符

换行符'\n'、回车符'\r'、空白符'\s'、制表位符'\t'等，

要用'\'转义或加r前缀统一转义

谢　谢