day6

一、今日内容

a. 约会对象预测练习

b. 特征预处理

c. 特征抽取

d. 朴素贝叶斯算法

二、 昨日复习

a. 模块

  i. import mio

  ii. import mio as a

  iii. from xxx import xxx

b. 异常

  i. 概念：会导致程序终止

  ii. 捕获异常

```
try:
    有可能发生异常的代码
except Exception as e:
    code
```

c. 人工智能历史

  i. 1950 ~ 1980： 简单的机械运动

  ii. 1980 ~ 2010：机器学习自然语言

  iii. 2010 ~ ： 深度学习处理图片

d. 机器学习概念

  i. 从历史数据中，分析获取规律，然后通过规律(模型)对结果进行预测

e. 数据

  i. 数据来源：BOSS 花钱购买 爬虫爬取 ....

  ii. 数据类型： 离散型数据 连续性数据

  iii. 数据格式： 数据集 ： 特征值 + 目标值

  iv. 分割数据集的方法： train_test_split

f. Knn算法

  i. 在一个和样本空间中，选取K个最相近的点，如果这K个点中，大部分是属于同一个类别，那么我也认为我是属于这个类别

  ii. 欧式距离公式：

iii. K值选取：选取奇数个数值。

iv. api：

1. __init__
2. fit()
3. predict()
4. score()

三、约会案例

相亲约会对象数据，这个样本时男士的数据，三个特征，玩游戏所消耗时间的百分比、每年获得的飞行常客里程数、每周消费的冰淇淋公升数。然后有一个所属类别，被女士评价的三个类别，不喜欢didnt、魅力一般small、极具魅力large也许也就是说飞行里程数对于结算结果或者说相亲结果影响较大，但是统计的人觉得这三个特征同等重要。

里程数	公升数	消耗时间比	评价
14488	7.153469	1.673904	smallDoses
26052	1.441871	0.805124	didntLike
75136	13.147394	0.428964	didntLike
38344	1.669788	0.134296	didntLike
72993	10.141740	1.032955	didntLike
35948	6.830792	1.213192	largeDoses
42666	13.276369	0.543880	largeDoses
67497	8.631577	0.749278	didntLike
35483	12.273169	1.508053	largeDoses
50242	3.723498	0.831917	didntLike

```python
# target: 预测 [[50000, 5, 0.5]] 女神对我的态度

# 导入pandas处理数据
import pandas as pd

# 定义目标值名称
target_names = ["didntLike", "smallDoses", "largeDoses"]


def main():

    # 1.获取原始数据集 pd.read_csv() 读取数据之后会返回一个DataFream数据类型
    dating = pd.read_csv("./dating.txt")

    # dataDream支持像字典一个读取数据，也支持切片操作
```

```
    print(dating)

    # 2. 确定特征值与目标值
    x = dating[["milage", "Liters", "Consumtime"]]
    y = dating["target"]

    print("x : ",x[:1])

    return 0

main()
```

四、特征预处理

4.1 概念：

将数字类型的数据，通过转换函数，将数据转换为适合机器学习的一组数据。

特征预处理： 归一化， 标准化

4.2 归一化

4.2.1 概念

概念： 将特征值数据，转换为 [0, 1]的范围之间。

4.2.2 公式

公式：

需要转换的特征值

$$X' = \frac{x - min}{max - min}$$

这个特征中的最小值

特征值中的最大值

将数据转换到需要的范围之间

$$X'' = X' * (mx - mi) + mi$$

需要转换的范围的最小值。

需要转换范围的最大值

4.2.3 归一化 API接口

```
# 导入特征预处理的api接口
from sklearn.preprocessing import MinMaxScaler
"""
MinMaxScaler类

__init__成员函数

函数原型:
    def __init__(self, feature_range=(0, 1));

函数参数:
    feature_range:需要转换的数据范围

fit成员函数

函数原型:
    fit(self, X, y=None);

函数功能:
    计算转换数据的标准。

函数参数:
    X: 需要计数转换数据的标准的世俗据

transform函数
函数原型:
    transform(self, X):

函数功能:
    通过给定的数据范围,对数据进行转换处理

函数参数:
    X: 需要转换的数据

返回值:
    转换数据的结果

"""

data = [[90, 2, 10, 40],
        [60, 4, 15, 45],
```

```
        [75, 3, 13, 46]]

convert = MinMaxScaler(feature_range=(0, 1))

# 制定转换数据的标准
convert.fit(data)
result = convert.transform(data)
print(result)


# target: 预测 [[50000, 5, 0.5]] 女神对我的态度

# 导入pandas处理数据
import pandas as pd

# 导入分割数据集的方法
from sklearn.model_selection import train_test_split
# 导入knn算法
from sklearn.neighbors import KNeighborsClassifier
# 导入归一化处理数据的傀儡
from sklearn.preprocessing import MinMaxScaler


# 定义目标值名称
target_names = [0, "didntLike", "smallDoses", "largeDoses"]


def main():

    # 1.获取原始数据集 pd.read_csv() 读取数据之后会返回一个DataFream数据类型
    dating = pd.read_csv("./dating.txt")

    # dataDream支持像字典一个读取数据，也支持切片操作

    # 2. 确定特征值与目标值
    x = dating[["milage", "Liters", "Consumtime"]]
    y = dating["target"]

    # 3. 划分数据集,将数据集划分为训练集和测试集
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,random_state

    # 归一化处理数据
    convert = MinMaxScaler()
```

```
    convert.fit(x_train)
    x_train = convert.transform(x_train)
    x_test = convert.transform(x_test)

    # 4. 实例化一个算法对象
    estimate = KNeighborsClassifier(n_neighbors=7)

    # 5. 拟合数据制作模型
    estimate.fit(x_train, y_train)

    # 6. 计算模型的准确度
    score = estimate.score(x_test, y_test)
    print("score : ", score)

    # 7. 预测模型的结果
    data = [[80000, 5, 0.5]]

    # 8 对预测数据进行归一化处理。

    # 对预测数据进行归一化处理
    data = convert.transform(data)
    y_predict = estimate.predict(data)
    print("模型的预测结果: ", target_names[int(y_predict)])
    return 0


main()
```

五、特征抽取

5.1 概念

将任意类型的数据(图片，字符串)，转换为适合机器学习的数字类型的特征

5.2 特征抽取原理

统计字符串中，每一个单词出现的次数，作为转换之后的特征值。

i. 制定转换数据的标准： 统计字符串中出现的单词，去除掉不重要的单词

ii. 统计出现的次数

```
data = ["life is short,i like python",
```

```
"life is too long,i dislike python"]

# 通过每一个单词出现的次数,作为特征值。

# 制定一个转换数据的标准
    [life,  is, short, like, python, too, long, dislike]
数据进行转换.
第一篇:  1  1   1  1  1  0  0   0
第二篇:  1  1   0  0  1  1  1   1
```

5.3 特征抽取的API接口

```
from sklearn.feature_extraction.text import CountVectorizer
"""
CountVectorizer类

__init__成员函数

函数原型:
    def __init__(self);

函数功能:
    初始化一个特征抽取的对象


fit成员函数

函数原型:
    fit(self, raw_documents, y=None);

函数功能
     统计字符串中出现的单词，去除掉不重要的单词

函数参数:
    raw_documents: 需要转换的文章


transform成员函数

函数原型:
    transform(self, raw_documents);
```

```
函数功能:
    对原始数据进行转换处理

函数返回值:
    sparse类型矩阵:  方便计算机存储

get_feature_names()函数
函数功能:
    返回制定的转换数据的标准

函数使用方式:
    obj.get_feature_names()

toarray()函数
函数功能:
    将数据转换为数组展示出来

使用方式:
    obj.toarray()
"""

data = ["life is short,i like python", "life is too long,i dislike python"]

# 实例化转换数据的对象
concert = CountVectorizer()

# 制定转换数据的标准
concert.fit(data)
print(concert.get_feature_names())

# 开始转换数据
result = concert.transform(data)
print(result.toarray())
```

六、中文分词 jieba 模块

6.1 jieba模块安装

i. windows + r ---> cmd + 回车

ii. pip install jieba

6.2 jieba模块使用

```
# jieba模块
import jieba  # 导入结巴模块
"""
jieba.cut函数

函数原型:
    cut(self, sentence);

函数功能:
    对中文文章进行分词操作

函数参数:
    sentence: 需要进行分词的中文语句

函数返回值:
    返回值:  生成器，如果需要看内部的内容，需要转换为列表操作的。
"""

# 原始数据
data = "新鲜草莓"

# 调用cut函数进行分词处理
result = list(jieba.cut(data))

print(result)
```

6.3 中文文章抽取

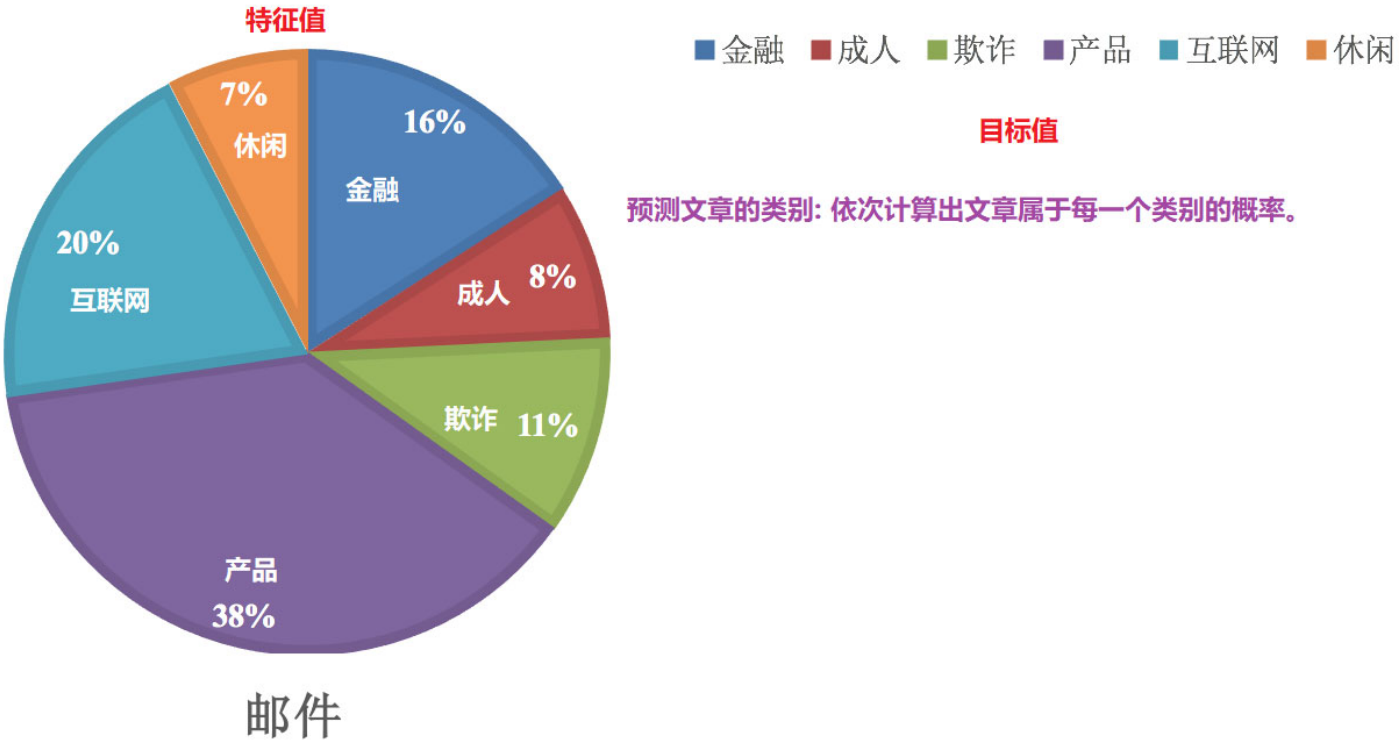
```
"""
今天很残酷，明天更残酷，后天很美好，但绝对大部分是死在明天晚上，所以每个人不要放弃今天。

我们看到的从很远星系来的光是在几百万年之前发出的，这样当我们看到宇宙时，我们是在看它的过去。

如果只用一种方式了解某样事物，你就不会真正了解它。了解事物真正含义的秘密取决于如何将其与我们所了
"""

# 导入结巴模块
import jieba
```

```python

# 导入特征抽取
from sklearn.feature_extraction.text import CountVectorizer


def cut_word():

    # 获取原始数据集
    c1 = "今天很残酷，明天更残酷，后天很美好，但绝对大部分是死在明天晚上，所以每个人不要放弃今天
    c2 = "我们看到的从很远星系来的光是在几百万年之前发出的，这样当我们看到宇宙时，我们是在看它的
    c3 = "如果只用一种方式了解某样事物，你就不会真正了解它。了解事物真正含义的秘密取决于如何将其

    # result的结果是一个列表。将列表里面的词语组成一个字符串. 列表里面的字符串需要使用 space隔
    result1 = list(jieba.cut(c1))
    result2 = list(jieba.cut(c2))
    result3 = list(jieba.cut(c3))

    ret1 = ""
    ret2 = ""
    ret3 = ""

    for i in result1:
        # 1.里面的每一个值，赋值给I
        ret1 += i
        ret1 += " "
    for i in result2:
        # 1.里面的每一个值，赋值给I
        ret2 += i
        ret2 += " "
    for i in result3:
        # 1.里面的每一个值，赋值给I
        ret3 += i
        ret3 += " "
    return ret1, ret2, ret3


# 特征抽取处理数据
def main():

    # 1.获取原始数据集
    ret1, ret2, ret3 = cut_word()

    # 2.实例化转换器对象
    concert = CountVectorizer()
```

```
    concert.fit([ret1, ret2, ret3])

    # 3.对数据进行转换
    result = concert.transform([ret1, ret2, ret3])
    print(concert.get_feature_names())
    print("result : ", result.toarray())


    return 0

main()
```

七、文章分类 -- 朴素贝叶斯算法

7.1 前言



7.2 概率

7.2.1 概念

一件事情发生的可能性，可能发生，也可能不发生。 记作 P(A)

7.2.2 概率计算

1. 联合概率： 不相关的事情同时发生的概率。
    a. 记作： P(A, B) = P(A) + P(B)
2. 条件概率：在某个条件的情况下，事件发生的概率

a. 记作： P(A | B)

样本数	职业	体型	女神是否喜欢
1	程序员	超重	不喜欢
2	产品	匀称	喜欢
3	程序员	匀称	喜欢
4	程序员	超重	喜欢
5	美工	匀称	不喜欢
6	美工	超重	不喜欢
7	产品	匀称	喜欢

1、女神喜欢的概率？

P(女神喜欢) = 喜欢 / 全部 = 4 / 7

2、职业是程序员并且体型匀称的概率？

P(程序员，匀称) = P(程序员) * P(匀称) = 12/ 49

3、在女神喜欢的条件下，职业是程序员的概率？

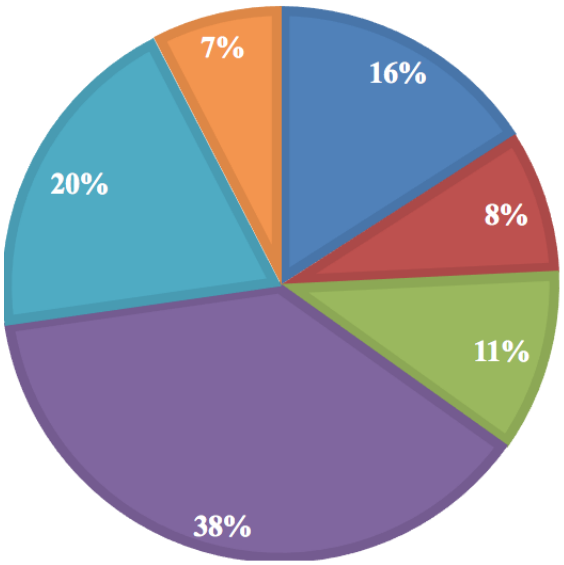
P(程序员|女神喜欢) = 1/ 2

4、在女神喜欢的条件下，职业是产品，体重是超重的概率？

P(产品，超重|女神喜欢) = 1/ 2 * 1/4 = 1/8

7.3 文章概率如何计算?

特征值



邮件

目标值

金融 成人 欺诈 产品 互联网 休闲

target: 计算邮件属于产品的概率?

P(产品|邮件) : 条件概率

特征抽取：将邮件转换为 特征词

P(产品 | word1, word2 .... wordN)

$$P(C|W) = \frac{P(W|C)P(C)}{P(W)}$$

类别 文章

P(word1, word2... wordN| 产品) * P(产品)

P(word1, word2 .... wordN)

P(word1|产品) * P(word2 | 产品) * ..... * P(wordN|产品) * P(产品)

P(word1) * P(word2) * ..... * P(wordN)

7.4 手动计算文章所属类别

训练集统计结果(指定统计词频)：

特征\统计	科技(30篇)	娱乐(60篇)	汇总（90篇）
“商场”	9	51	60
“影院”	8	56	64
“支付宝”	20	15	35
“云计算”	63	0	63
汇总(求和)	100	121	221

现有一篇被预测文档：出现了影院，支付宝，云计算，计算属于科技、娱乐的类别概率？

P(影院|科技) * P(支付宝|科技) * P(云计算| 科技) * P(科技)

P(影院) * P(支付宝) * P(云计算)

(8 / 100) * (20 / 100) * (63 / 100) * (30 / 90)

(64/221) * (35 / 221) * (63/ 221)

≈ 0.00335664 / 0.0130740983 ≈ 25.67%

target: 计算文章属于科技的类别是多少?

P(科技|文章)

|| 对文章进行特征抽取

P(科技 |影院, 支付宝,云计算)

|| 贝叶斯公式

P(影院, 支付宝, 云计算| 科技) * P(科技)

P(影院, 支付宝, 云计算)

||

P(影院|科技) * P(支付宝|科技) * P(云计算| 科技) * P(科技)

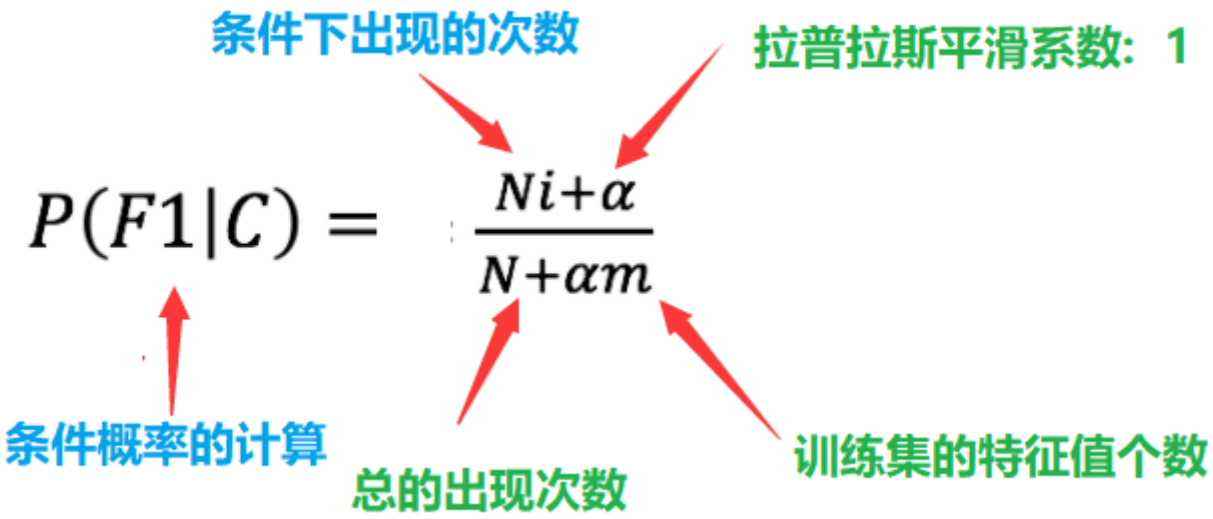
P(影院) * P(支付宝) * P(云计算)

7.5 拉普拉斯平滑系数

i. 目的:防止计算的概率为 0

ii. 拉普拉斯平滑系数: 添加在条件概率中的.

iii. 拉普拉斯平滑系数公式:



7.6 朴素贝叶斯算法API接口

```
# 朴素贝叶斯算法API接口
from sklearn.naive_bayes import MultinomialNB
"""
MultinomialNB成员函数

__init__成员函数:

函数原型:
    __init__(self, *, alpha=1.0);

函数参数:
    alpha: 拉普拉斯平滑系数


fit成员函数

函数原型:
    fit(self, X, y);

函数功能:
    通过训练集数据,制作模型

函数参数:
    X: 训练集的特征值
    y: 训练集的目标值


predict(self,x): 通过数据预测模型的标签值
```

```
score(self, x, y):计算模型的准确度
"""
```

八、新闻分类案例

20个新闻组

20个新闻组数据集

20个新闻组数据集是大约20,000个新闻组文档的集合，平均分布在20个不同的新闻组中。据我所知，它最初是由Ken Lang收集的，可能是因为他的*Newsweeder：学习过滤网络新闻纸*，尽管他没有明确地提到这个集合。这20个新闻组集合已经成为机器学习技术的文本应用中的实验流行数据集，例如文本分类和文本聚类。

组织

数据被组织成20个不同的新闻组，每个新闻组对应不同的主题。一些新闻组彼此之间关系密切（例如**comp.sys.ibm.pc.hardware / comp.sys.mac.hardware**），而另一些新闻组则非常不相关（例如**misc.forsale / soc.religion.christian**）。以下是根据主题划分的20个新闻组列表（或多或少）：

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

```python
# 导入新闻网站的数据
from sklearn.datasets import fetch_20newsgroups
# 导入分割数据集的方法
from sklearn.model_selection import train_test_split
# 导入特征抽取，对数据进行处理
from sklearn.feature_extraction.text import CountVectorizer
# 导入朴素贝叶斯算法
from sklearn.naive_bayes import MultinomialNB



def main():

    # 1.获取到原始数据集
    news = fetch_20newsgroups(subset="all") # 训练集数据与测试集数据全都获取

    # 2.确定特征值与目标值
    x = news.data
    y = news.target

    # 3.划分数据集,将数据集划分为训练集和测试集
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)

    # 4.实例化一个转换数据的对象
    concert = CountVectorizer()
```

```
    # 5.制定转换数据的标准
    concert.fit(x_train)

    # 6.开始转换数据
    x_train = concert.transform(x_train)
    x_test = concert.transform(x_test)

    # 7.实例化朴素贝叶斯算法对象
    estimate = MultinomialNB()
    estimate.fit(x_train, y_train)

    score = estimate.score(x_test, y_test)
    print(score)

    # 8.对数据进行预测
    data = """Amid the rapprochement momentum in the Middle East and the US' strategic c

Blinken's trip, from June 6 to 8, comes after a visit by White House national security a

According to the US State Department, Blinken will meet with Saudi officials to "discuss

The top US diplomat will also attend the Gulf Cooperation Council (GCC) talks during his

Before Blinken's visit, Saudi Arabia's foreign minister Faisal bin Farhan met with his I

In April, Saudi Arabia and Iran formally restored diplomatic ties after a seven-year rif

With the rapprochement process in the Middle East accelerating, observers are not optimi

Liu Zhongmin, a professor with the Middle East Studies Institute of Shanghai Internatior

On the one hand, the US is shifting its strategic focus to the Asia-Pacific region to cc

The US cannot stop the trend of increasing autonomy in the Middle East, so it is trying

Li Haidong, a professor at the China Foreign Affairs University, told the Global Times c

In the process of adapting to the new situation in the Middle East, the US would try to

While the US is less likely to play a constructive role in the Middle East, its disrupti

The US is likely to take stock of the situation and become more involved in the region i

```

```
The oil-producing countries of the Middle East seem to have woken up to the risks, espec

On June 2, Saudi minister of energy Prince Abdulaziz bin Salman met Zhang Jianhua, admir

Middle East countries faced a historic opportunity to liquidate the power of the US and

Chinese Foreign Ministry spokesperson Wang Wenbin said on Monday that China supports reg

As a good friend of regional countries, China will continue to play an active and constr

    data = concert.transform([data])
    result = estimate.predict(data)

    print("文章的类别:  ", news.target_names[int(result)])
    return 0


main()
```