机器学习

第6次实验：使用朴素贝叶斯对垃圾邮件分类

实验目的

01 会用Python创建朴素贝叶斯模型

02 使用朴素贝叶斯模型对垃圾邮件分类

03 会把文本文件变成向量

04 会用评价朴素贝叶斯模型的分类效果

实验内容

- 把给定的数据集message.csv拆分成训练集和测试集，使用sklearn. naive_bayes.MultinomialNB类创建一个朴素贝叶斯模型，使用训练数据训练出一个预测模型，然后用预测模型对测试集中数据进行分类，评价模型的分类效果

- message.csv数据集中包含大量的短信，每行数据包括2个字段：短信内容，短信类别（1或者0）,短信类别为1的是垃圾邮件

- MultinomialNB对象的α属性，可以用于设置或获取相应的平滑参数值。

实验内容

1

读取CSV文件，将数据集按3:1的比例拆分成训练集合测试集

```
split_ratio = 0.75
training_data = []
testing_data = []
np.random.seed(0)
```

将文本拆分成单词函数

```
def tokenize(message):
    message = message.lower()
    all_words = re.findall("[a-z0-9']+", message)
    return set(all_words)
```

实验内容

2 构建词汇表，形成特征矩阵和分类矩阵

```
def generateMat(data):
    num_samples = len(data)
    feature = np.zeros((num_samples, num_features))
    classify = np.zeros(num_samples)
    for i in range(num_samples):
        data_row = data[i]
        classify[i] = data_row[1]
        for word in data_row[0]:
            if word in word_dict:
                feature[i][word_dict.index(word)] = 1
    return feature, classify
```

实验内容

3 根据训练数据生成特征矩阵和分类矩阵，显示训练矩阵特征维度

4 根据测试数据生成特征矩阵和分类矩阵，显示测试矩阵特征维度

5 用训练集训练朴素贝叶斯模型

6 用测试集进行预测

实验内容

7 计算并显示模型的准确率、精度、召回率和F1值

```
TN = FP = TP = FN = 0
for i in range(len(predict_classify)):
    if testing_classify[i] == 0 and predict_classify[i] == 0:
        TN += 1
    if testing_classify[i] == 0 and predict_classify[i] == 1:
        FP += 1
    if testing_classify[i] == 1 and predict_classify[i] == 1:
        TP += 1
    if testing_classify[i] == 1 and predict_classify[i] == 0:
        FN += 1
p = TP / (TP + FP)
r = TP / (TP + FN)
```

执行结果图例

```
1.读取csv文件数据并拆分成训练数据和测试数据.....
2.构造词汇表，并形成Feature矩阵和Classify矩阵....
3.根据训练数据，生成feature矩阵和classify矩.....
训练矩阵特征维度：（2418，3746）
4.根据测试数据，生成feature矩阵和classify矩.....
测试矩阵特征维度：（850，3746）
5.训练朴素贝叶斯模型.....
6.用测试集预测.....
7.评价模型，计算准确率、精度、召回率和F1值.....
Accuracy： 0.9423529411764706
Precision： 0.8924731182795699
Recall： 0.6803278688524459
F1 Score: 0.7720930232558139
```