**《大数据分析与处理》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2019级计算机科学与技术（卓越）** | | | **姓名** | **李燕琴 李放** |
| **实验题目** | **实验三 文本分类算法** | | | | | |
| **实验时间** | **2022年5月5日** | | **实验地点** | **DS3 302** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 √设计性 □综合性** | | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 一、实验目的   * 1.掌握数据预处理的方法，对训练集数据进行预处理； * 2.掌握文本建模的方法，对语料库的文档进行建模； * 3.掌握分类算法的原理，基于有监督的机器学习方法，训练文本分类器； * 4.利用学习的文本分类器，对未知文本进行分类判别； * 5.掌握评价分类器性能的评估方法。 | | | | | | |
| 三、实验过程或算法（源程序）  本次实验内容为进行文本分类算法的实现，具体为**对购物评论进行正向负向的分类**，实验选用数据集为online\_shopping\_10\_cats数据集。数据集中包含三个字段：1.cat表示类别，共有10个类别；2.label表示评论分类，1为正向评论，0为负向评论；3.review表示评论内容。数据集中正、负向评论各约 3 万条，请同学们先将数据集按照train\_size:test\_size=6:4的比例划分为训练集和测试集，再进行文本分类实验。  实验过程如下：   1. **文本预处理**   文本预处理的任务是去除文本的噪声信息，例如HTML标签，文本格式转换，检测语句边界等，最终将文本处理为结构化的数据。本实验选用的数据集已被预处理过。   1. **中文分词**   由于中文词语没有空格分割，需要首先使用中文分词器为文本分词，并去除停用词。停用词是指对文本内容影响较弱的词汇，需要自行设置停用词表，然后按表将训练文本中的停用词剔除。目前常用jieba分词器进行中文分词。   1. **文本表示**   文本表示是指将文本表示成向量的形式。本次实验请分别用**LDA表示**和**TF-IDF表示**两种表示方法生成文本表示向量，并通过对比不同方法下的实验效果分析方法的优劣。  **LDA主题模型**为生成式模型，假设待建模的文本数据为X，其分类主题标签为Y，LDA模型的本质是联合概率分布P(X,Y)的生成过程。LDA模型原理如公式(1)所示，  (1)  其中，表示文档数据形成的矩阵，为文档总数，为语料库中单词总数，表示词在文档中出现的频率；表示document - topic矩阵，为主题总数，每列表示一个主题，每个行向量表示文档的主题分布；表示topic - term矩阵，每行为一个分布，表示主题在语料库上的多项式分布。  LDA假设文档是由多个主题的混合来产生的，每个文档的生成过程如下：   1. 从全局的泊松分布参数为的分布中生成一个文档的长度，表示 topic-word密度，越高，主题包含的单词更多，反之包含的单词更少； 2. 从全局的狄利克雷参数为的分布中生成当前文档的主题分布，表示 document-topic密度，越高，文档包含的主题更多，反之包含的主题更少； 3. 从主题的多项式分布中取样生成文档第个词的主题； 4. 从狄利克雷分布中取样生成主题的词语分布； 5. 从词语多项式分布中采样最终生成词语；   训练过程为吉布斯采样过程，详细如下：  吉布斯采样 (Gibbs Sampling) 首先选取概率向量的一个维度，给定其他维度的变量值当前维度的值，不断收敛来输出待估计的参数。具体地   1. 随机给每一篇文档的每一个词，随机分配主题编号； 2. 统计每个主题下出现词的数量，以及每个文档中出现主题中的词的数量； 3. 每次排除当前词的主题分布，根据其他所有词的主题分类，来估计当前词分配到各个主题的概率，即计算。得到当前词属于所有主题的概率分布后，重新为词采样一个新的主题。用同样的方法不断更新的下一个词的主题，直到每个文档下的主题分布和每个主题下的词分布收敛； 4. 最后输出待估计参数和，每个单词的主题也可以得到。   在大量的迭代后，主题分布和词分布都比较稳定也比较好了，LDA模型收敛。  LDA模型的实现可借助Jieba分词工具和gensim工具包，具体方法请自行网上查阅。  **TF-IDF**方法本质为根据字词的在文本中出现的次数和在整个语料中出现的文档频率来计算一个字词在整个语料中的重要程度。优点在于能过滤掉一些常见的却无关紧要的词语，并保留影响整个文本的重要字词。计算方法如公式(2)所示,  (2)  (3)  (4)  其中，表示特征词对文档的重要程度，值越大越重要；为词频，表示特征词在文档中出现的频率，计算公式如公式(3)所示，其中，为特征词在文档中出现次数，为文档中各特征词出现总数；表示倒文本词频，表示某个特征词在整个语料所有文档中出现的次数，计算公式如公式(4)所示，其中，表示语料中文本的总数，表示文本中包含特征词的数量。  TF-IDF模型可自己实现，也可使用scikit-learn提供的文本分析工具包实现，具体方法请自行网上查阅。   1. **文本分类器**   用TF-IDF或者LDA构造出词向量之后，直接调用下游的scikit-learn机器学习工具包中分类器（如逻辑回归、支持向量机、神经网络、决策树、XGBoost等）进行分类。   1. **评价指标**   对测试集的分类结果利用正确率和召回率进行分析评价：计算正负类分别的准确率、召回率，计算总体准确率和召回率，以及总体的*F*1-score，这里*F*1-score公式如(5)所示，  (5)  其中，为准确率，为召回率。 | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程  1、LDA构造的词向量  根据数据，构造LDA模型时，指定10个主题，如图 1所示。并得出每个句子每个主题的概率如图 2所示。得到(62773, 10)维度的X。  图 1 LDA表示  图 2 LDA词向量  2、TF-IDF构造的词向量  根据TF-IDF的计算，取排名前2000的词向量表示作为该句的向量表示，如图 3所示。得到(62773, 2000)维度的X。  图 3 TF-IDF表示  图 4 TF-IDF的稀疏表示  3、模型运行结果  由下表可以看出：   * 对于LDA来说，XGBoost结合了集成学习的方法，表现效果最好。其次是支持向量机。 * 对于TF-IDF来说，维数较高，表征的信息更多，逻辑回归、支持向量机、神经网络的效果都很好。  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Model | LDA | | | TF-IDF | | | | P | R | F1 | P | R | F1 | | 逻辑回归 | 0.73 | 0.73 | 0.73 | 0.82 | 0.79 | 0.78 | | 支持向量机 | 0.75 | 0.75 | 0.75 | 0.82 | 0.79 | 0.78 | | 神经网络 | 0.73 | 0.73 | 0.73 | 0.82 | 0.79 | 0.78 | | 决策树 | 0.70 | 0.70 | 0.70 | 0.80 | 0.77 | 0.76 | | XGBoost | 0.77 | 0.77 | 0.77 | 0.78 | 0.70 | 0.67 |  |  | | --- | | 4、源代码 | | import pandas as pd  import jieba  import tqdm  import numpy as np  from gensim import corpora  from gensim.models import LdaModel,TfidfModel  from sklearn.model\_selection import train\_test\_split, cross\_val\_score  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.metrics import classification\_report  from sklearn.svm import LinearSVC  from sklearn.linear\_model import LogisticRegression  from sklearn.neural\_network import MLPClassifier  from sklearn.tree import DecisionTreeClassifier  from xgboost import XGBClassifier  '''停用词'''  # 中英文特殊符号  bd = '~`!@#$%^&\*(){}[]:;"\'|\,.<>?/·！（）：；”‘’《》，。、'  bd = [bd[i] for i in range(len(bd))]  def load\_dictionary(path):      stopwords = []      with open(path,encoding='utf-8') as src:          for word in src:              word=word.strip()              stopwords.append(word)      stopwords +=bd      return stopwords  '''读入数据'''  data\_path = "./online\_shopping\_10\_cats.csv"  stop\_path = "./stopwords-master/baidu\_stopwords.txt"  data = pd.read\_csv(data\_path)  # 去除空review，特殊字符处理  data.dropna(inplace=True)  data.index = list(range(len(data)))  # 文本分词  sens = data['review'].apply(lambda x:jieba.lcut(x))  # 去除停用词  stopwords = load\_dictionary(stop\_path)  texts = []  for senl in tqdm.tqdm(sens):      senl = [sen for sen in senl if sen not in stopwords]      texts.append(senl)  Y = list(data['label'])  '''基本情况+建立字典'''  num\_topics = len(set(data['cat']))  dictionary = corpora.Dictionary(texts)  corpus = [dictionary.doc2bow(text) for text in texts] # 利用语料特征，为每一句话，建立稀疏向量（这里是bow向量）。对应特征（单词）出现的次数  n\_samples = len(texts)  max\_sen\_len = max([ len(text) for text in texts])  max\_cor\_len = max([ len(cor) for cor in corpus])  len\_dict = len(dictionary)  print(f"n\_samples = {n\_samples}")  print(f"max\_sen\_len = {max\_sen\_len}")  print(f"max\_cor\_len = {max\_cor\_len}")  print(f"len\_dict = {len\_dict}")  '''构造X'''  def get\_LDA\_X():      myLdaModel = LdaModel(corpus, id2word=dictionary, iterations=50, num\_topics=num\_topics)      myLdaModel.print\_topic(1,topn=10)      X = np.zeros((n\_samples, num\_topics))      for i in range(len(corpus)):          for j,k in myLdaModel[corpus[i]]:              X[i,j] = k      print("LDA构造的词向量: ",X.shape)      return X  def get\_TFIDF\_X():      vectorizer = TfidfVectorizer(stop\_words = stopwords,max\_features=2000)      X = vectorizer.fit\_transform(data['review'])      X = X.todense()      print("TF-IDF构造的词向量: ",X.shape)      return X  Xd = {      "LDA":get\_LDA\_X(),      "TF-IDF":get\_TFIDF\_X(),  }  '''模型训练'''  for k in Xd:      if k=="LDA":          continue      X = Xd[k]      print("="\*20+k+"="\*20)      model\_list = {          "逻辑回归":LogisticRegression(),          "支持向量机":LinearSVC(),          "神经网络":MLPClassifier(solver='adam', # solver 学习器， alpha L2惩罚参数                              hidden\_layer\_sizes=(X.shape[1]//2, 2),random\_state=1,max\_iter=1),          "决策树":DecisionTreeClassifier(),          "XGBoost":XGBClassifier(),      }      assert len(X)==len(Y)      X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.4, random\_state=42)      for key in tqdm.tqdm(model\_list):          model = model\_list[key]          print("="\*20+key+"="\*20)          k = 5          # print("%d折交叉训练结果："%(k),cross\_val\_score(model, X, Y, cv=k))          model.fit(X\_train,Y\_train)          Y\_pred = model.predict(X\_test)          print("测试结果：")          print(classification\_report(Y\_test,Y\_pred)) | | | | | | | |

备注：

1、教师在布置需撰写实验报告的实验前，应先将报告书上的“实验题目”、“实验性质”、“实验目的”、“实验项目内容”等项目填写完成，然后再下发给学生。

2、教师在布置需撰写报告的实验项目时，应告知学生提交实验报告的最后期限。

3、学生应按照要求正确地撰写实验报告：

* 1. 在实验报告上正确地填写“实验时间”、“实验地点”等栏目。
  2. 将实验所涉及的源程序文件内容（实验操作步骤或者算法）填写在“实验过程或算法（源程序）”栏目中。
  3. 将实验所涉及源程序调试过程（输入数据和输出结果）或者实验的分析内容填写在“实验结果及分析和（或）源程序调试过程”栏目中。
  4. 在实验报告页脚的“报告创建时间：”处插入完成实验报告时的日期和时间。
  5. 学生将每个实验完成后，按实验要求的文件名通过网络提交（上载）到指定的服务器所规定的共享文件夹中。每个实验一个电子文档，如果实验中有多个电子文档（如源程序或图形等），则用WinRAR压缩成一个压缩包文档提交，压缩包文件名同实验报告文件名（见下条）。
  6. 提交的实验报告电子文档命名为：“年级（两位数字不要“级”字）专业（缩写：计算机科学与技术专业（计科）、网络工程专业（网络）、信息安全专业（信息）、物联网工程（物联网））班级（两位数字）学号（八位数字）姓名实验序号（一位数字）．doc。如学号为20115676、年级为2011级、专业为“计算机科学与技术”专业、班级为“02班”、姓名为“王宇”的学生，完成的第一次实验命名为： 11计科02班20115676王宇1．Doc，以后几次实验的报告名称以此类推。

4、教师（或助教）在评价学生实验时，应根据其提交的其他实验相关资料（例如源程序文件等）对实验报告进行仔细评价。评价后应完成的项目有:

1. 在“成绩”栏中填写实验成绩。每个项目的实验成绩按照五级制（优、良、中、及格、不及格）方式评分，实验总成绩则通过计算每个项目得分的平均值获得（平均值计算时需将五级制转换为百分制优=95、良=85、中=75、及格=65、不及格=55）。
2. 在“教师评价”栏中用符号标注评价项目结果（用√表示正确，用×表示错误，用≈表示 半对半错）。
3. 在“教师评价”栏中“评价教师签名”填写评价教师（或助教）姓名。将评价后的实验报告转换为PDF格式文件归档。
4. 课程实验环节结束后，任课教师将自己教学班的实验报告文件夹进行清理。在提交文件夹中，文件总数为实验次数×教学班学生人数（如，教学班人数为90人，实验项目为5，其文件数为：90×5=450）。任课教师一定要认真清理，总数相符，否则学生该实验项目不能得分。最后将学生提交的实验报告刻光盘连同实验成绩一起放入试卷袋存档。