README

代码文件解释

classify文件夹下均是分类相关文件包括

- 模型

  bayes.py--朴素贝叶斯分类

  NN_classify.py--神经网络

  minibatch_classify.py--小批量梯度下降神经网络

  RF_classification.py--随机森林

  enknn_classify.py--集成KNN

- 词向量

  read_classification.py--读取文件

  d2v_classification.py--doc2vec

  tfidf_classification.py--生成tfidf矩阵

  w2c.py--word2vec

regression文件夹下均是和回归相关文件包括

- 模型

  NN_reg.py--神经网络

  mini_batch_reg.py--小批量梯度下降的神经网络

  RF_regression.py--随机森林

  enknn_reg.py--集成KNN

- 词向量

  read_regression.py--读取文件

  d2v_regression.py--doc2vec

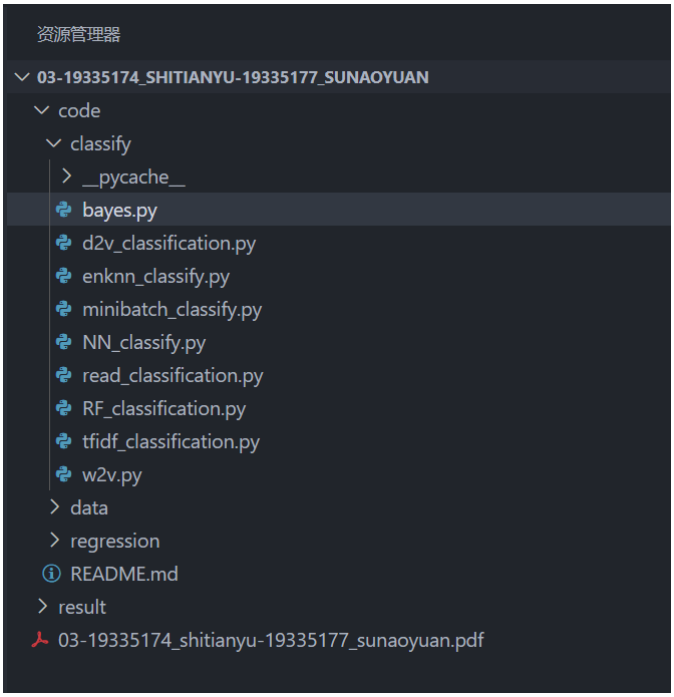
  tfidf_regression.py--生成tfidf矩阵

最优Private结果复现方法

分类

分类的最优Private结果是利用朴素贝叶斯分类器和TFIDF向量得到

所有模型和生成向量文件均在classify文件夹下



复现步骤如下:

- 运行**tfidf_classification.py**，在classify文件夹下生成

  train_vectors_TFIDF.npy；train_labels_TFIDF.npy；valid_vectors_TFIDF.npy；valid_labels_TFIDF.npy；test_vectors_TFIDF.npy共5个文件

- 运行**bayes.py**得到验证集分类正确率和文件bayes.csv

回归

回归的最优Private结果是利用小批量梯度下降的神经网络实现，词向量选择doc2vec

doc2vec参数设置:

```
vector_size=100, min_count=2, epochs = 20,workers = 4, window = 2
```

神经网络参数设置:

```
各层节点数
n0 = train_vectors.shape[1]
n1 = 50
n2 = 10
n3 = 1
学习率 0.01
迭代次数 8000
random_state = 10000
lamd = 0.01
激活函数LeakyRelu
batchsize = 256
```

复现步骤

- 运行regression文件夹下的d2v_regression.py生成

  train_vectors.npy;

  train_rating.npy;

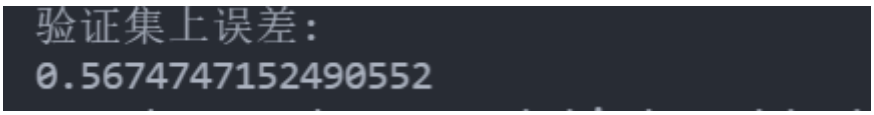
  valid_vectors.npy;

valid_rating.npy

test_id.npy

test_vectors.npy

- 运行regression文件夹下的mini_batch_reg.py文件生成csv文件



PS：因Doc2Vec结果每次都不同，且神经网络输出结果不稳定，回归的结果可能不能完全一致。分类结果是可以的。