# **2019年3月13日**

# （https://blog.csdn.net/YUNFONG\_Liao/article/details/87970959）

（http://www.medsci.cn/article/show\_article.do?id=bde1e582726）

# **[MRI] 核磁共振T1和T2图像**

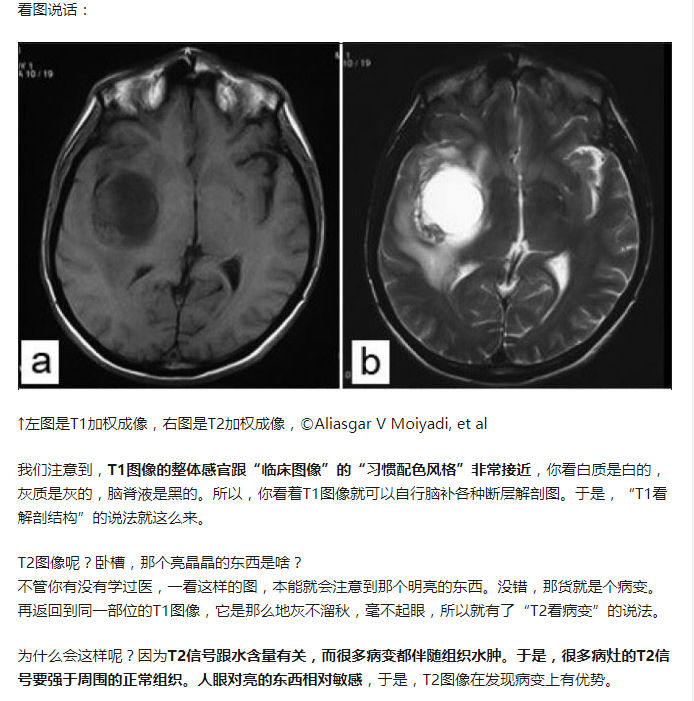
### 1、MRI的优点：无曝光，无入侵，无障碍摄影，强烈的组织对比度，通过改变拍摄时的频率，可以在同一位置进行多向拍摄并拍摄不同的信息，拍摄期间不受骨骼瑕疵的影响

### MRI的缺点：拍摄时间长，患者在拍摄期间移动时容易失败，设备稍窄，设备非常昂贵，检查声音大

### 2、关于T1加权图像和T2加权图像

### 在MRI拍摄中，通常会拍摄T1高光图像和T2高光图像的两种类图像作为一组。 在这两种类型的图像中，身体中强调了不同的组织，因此可以从每个图像中识别每个组织

1. T1观察解剖结构较好，T2显示组织病变较好。



4、\*（万能）医学缺点分析：在采集数据的过程中，由于患者肢体的移动、体位的不固定、身体相对于影像设备的位置不确定等因素，在所得的MR图像中往往含有伪差图像；成像设备的局限性和组织的續动会导致噪声和伪影、病变组织的病变边缘不明确、局部体效应导致组织边缘模糊、不均匀的磁场和组织器官等；

# **2019年3月16日**

\*初步代码工程第一次汇总：

数据集处理（D:\pycharm项目\pyradiomics\pydicom\）

[Lianxi.py----lianxi8.py]

配准数据集（E:\csp\）

依托于ants，主要进行仿射变换，在控制端运行bat格式文件

用于检查配准是否正确（D:\VS2010 Code\配准例子\配准例子.sln）

C++语言用于查看仿射变换后生成的nii图像与原图是否有相同的原点坐标

特征提取与训练（D:\pycharm项目\pyradiomics\）

1. py---5.py]

1.py进行特征查看；3.py主要进行svm训练；特征提取参数文件为（D:\pycharm项目\pyradiomics\examples\exampleSettings\Params.yaml）

批量操作（D:\pycharm项目\pyradiomics\examples\batchprocessing.py）

主要路径：(D:\pycharm项目\pyradiomics\examples\)

1. radiomics\_features.csv[提取特征表格]
2. Label.csv[label表格]
3. Lujing.csv[批量操作路径表格]

...............

# **2019年3月17日**

1、iris数据集学习：

因为后期要对自己数据集进行训练，而iris数据集形式非常有利于sklear中各种函数处理，所以学习此数据集，把自己的数据集变为iris形式。

所用代码3.py（D:\pycharm项目\pyradiomics\3.py）

iris数据集介绍参考：

<https://blog.csdn.net/brooknew/article/details/80915424>

https://blog.csdn.net/long\_long666/article/details/80506769

<https://blog.csdn.net/java1573/article/details/78865495>

<https://blog.csdn.net/u012679707/article/details/80511968>

清楚介绍

https://blog.csdn.net/qq\_36142114/article/details/80441373

数据转换参考：

https://blog.csdn.net/zhubao124/article/details/80719306

<https://blog.csdn.net/tymatlab/article/details/79009618>

<https://blog.csdn.net/weixin_39223665/article/details/81268741>

<https://blog.csdn.net/shengyingpo/article/details/52860537>

<https://blog.csdn.net/qq_35576544/article/details/79616071>

2、批量提取特征有个错误（UnicodeDecodeError: 'gbk' codec can't decode byte 0xda in position 28: illegal multibyte sequence）：自己建立的.csv路径表格用不了，编码错误，可能是WPS的问题。解决办法：此电脑桌面有个test.csv文件，可以在这个文件上填写自己的路径，然后再将改过后的.csv文件送入D:\pycharm项目\pyradiomics\examples\batchprocessing.py里运行。

3、整个代码工程后期主要是：batchprocessing.py和3.py，今天初步跑通实验。

4、API(Application Programming Interface,应用程序编程接口)

<https://blog.csdn.net/cumtdeyurenjie/article/details/80211896>

**2019年3月18日**

1. numpy中linspace（）简单用法：np.linspace(1,10,10)--即在1到10之间返回返回10个数，这10个数是等差数列，即输出--[ 1. 2. 3. 4. 5. 6. 7. 8. 9.10.]。并且结果可以看出，在默认情况下，linspace函数可以生成元素为50的等间隔数列。而前两个参数分别是数列的开头与结尾。如果写入第三个参数，可以制定数列的元素个数。

<https://blog.csdn.net/grey_csdn/article/details/54561796>

1. sklearn-GridSearchCV,CV调节超参使用方法（网格搜索算法）：

关于这个的学习是一个长期的过程！

**2019年3月19日**

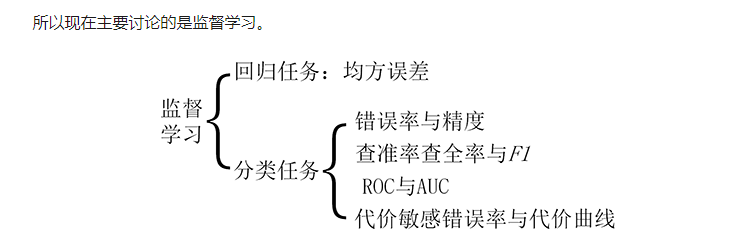
1. 机器学习模型性能评估

性能度量：

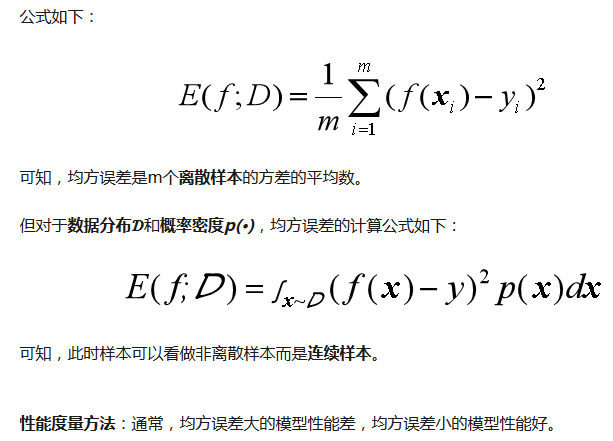
参考（https://blog.csdn.net/batuwuhanpei/article/details/51884351

<https://zhuanlan.zhihu.com/p/28448682>）

西瓜书第二章



1. 回归任务的性能度量——均方误差：



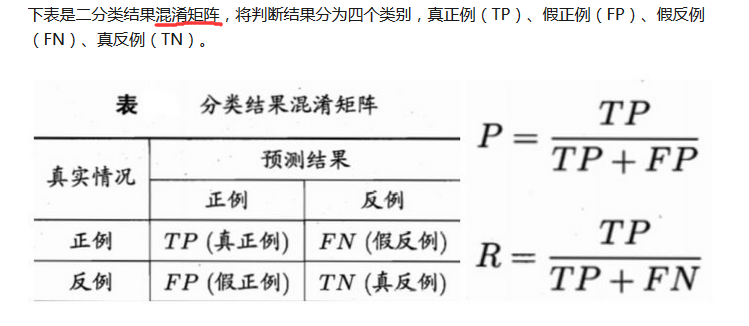
1. 分类任务的性能度量1——错误率与精度：

错误率：分类错误的样本占样本总数的比例。

精度：分类正确的样本占样本总数的比例。

性能度量方法：通常，错误率低精度高的模型性能好，错误率高精度低的模型性能差。错误率与精度反应的是分类任务模型判断正确与否的能力。

分类任务的性能度量2——查准率、查全率与F1：



查准率：【真正例样本数】与【预测结果是正例的样本数】的比值。

查全率：【真正例样本数】与【真实情况是正例的样本数】的比值。

一般来说，查准率高时，查全率偏低；查全率高时，查准率偏低。通常只在一些简单任务中，查准率和查全率都偏高。

分类任务的性能度量3——ROC与AUC

分类任务的性能度量4——代价敏感错误率与代价曲线

**2019年3月20日**

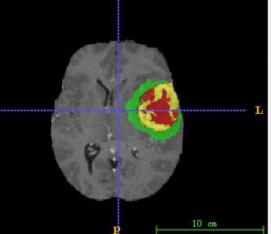
组会

BRATS 2017 数据集

红色是坏死，绿色是水肿，黄色是增强。

非增强和坏死两个类合并为一个类，所有现在的数据都是四个类。

label分别是0,1,2,4。



**2019年3月21日-----3月24日 外出**

**2019年3月27日**

1. 数据集划分：

from sklearn.model\_selection import train\_test\_split

<https://blog.csdn.net/qq_36955294/article/details/80135797>

2、交叉验证（****Cross Validation****）

主要是K折交叉验证：

将数据集（其实是训练集，我们暂且称为数据集吧，假设含有N个样本）分成K份（每份含有N/K个样本），选择其中1份作为测试集，另外K-1份作为训练集。这样的话，测试集就有K种情况。

 在每种情况中，用训练集训练模型，用测试集测试模型，计算模型的泛化误差（暂且这么叫吧）。

 将K种情况下，模型的泛化误差取均值，得到模型最终的泛化误差。

**2019年3月29日**

1、西瓜书模型评估：<https://zhuanlan.zhihu.com/p/28482121>

<http://alexkong.net/2013/06/introduction-to-auc-and-roc/> 主要学习ROC曲线，t检验

1. 归一化与标准化： <https://en.wikipedia.org/wiki/Feature_scaling>

<https://blog.csdn.net/pipisorry/article/details/52247379>（清楚）

https://en.wikipedia.org/wiki/Normalization\_(statistics)

<https://blog.csdn.net/dingxl555/article/details/82986362?utm_source=blogxgwz0>

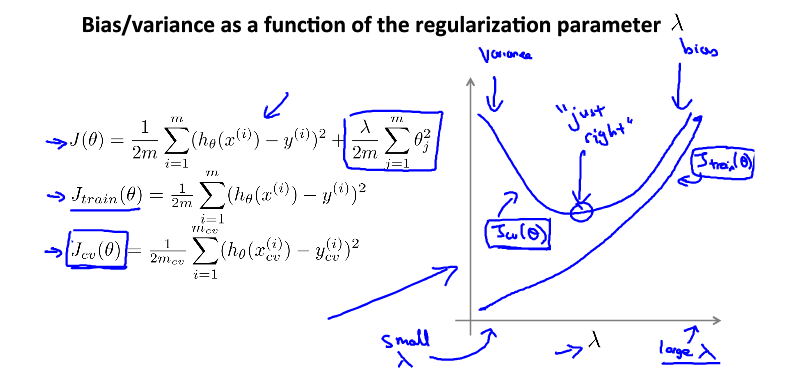
归一化与标准化不是一回事，归一化是标准化的方法之一，我的实验中用的是z-score 标准化（每个特征中的数值平均变为0、标准差变为1，结合Python中sklearn-preprocessing理解）

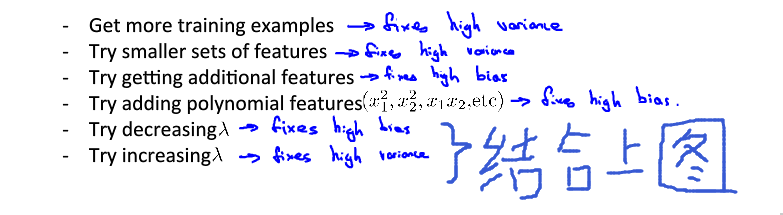
1. sklearn-preprocessi <https://blog.csdn.net/weixin_40807247/article/details/82793220>

<https://www.zhihu.com/question/28641663/answer/139203996>

**2019年4月1日**

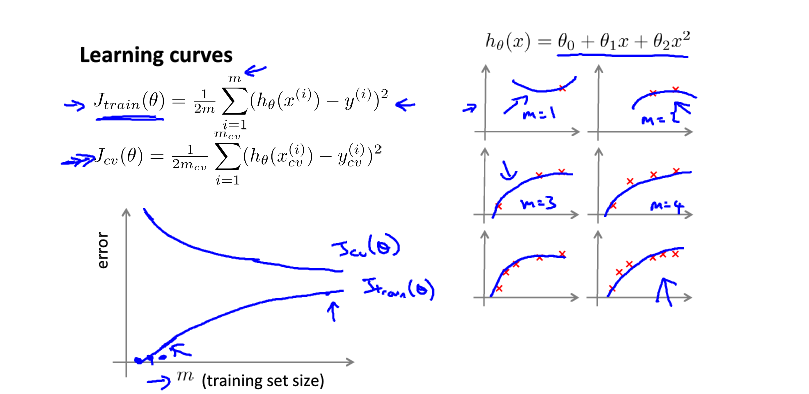
λ的影响



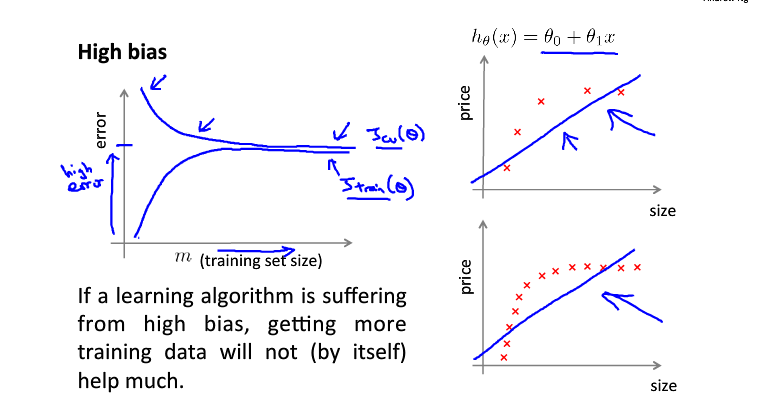


学习曲线

①模型‘正好’时

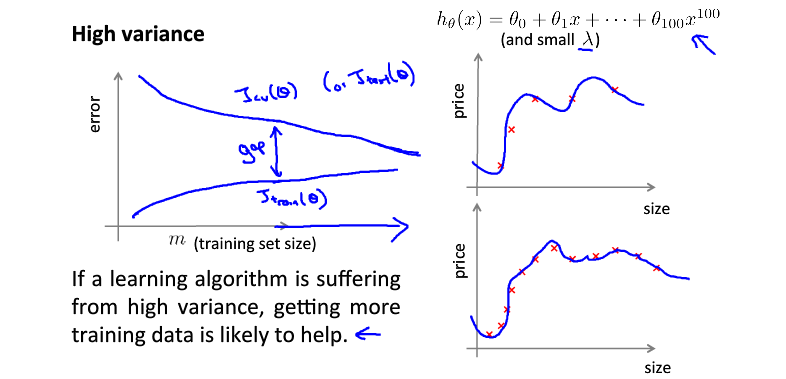


②高偏差



当处于高偏差时，加大数据并没有效果。注意此时error很大

③高方差

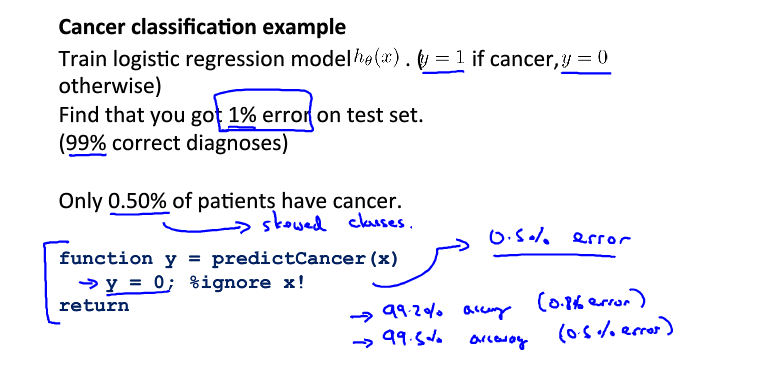


处于高方差时，加大数据可能有效果。注意此时训练集误差与测试集误差之间数值‘距离’大。

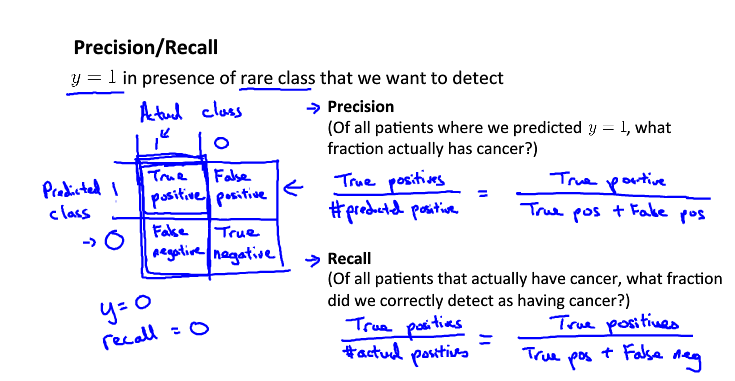
**2019年4月2日**

1、查准率、查全率与F1（F）理解：

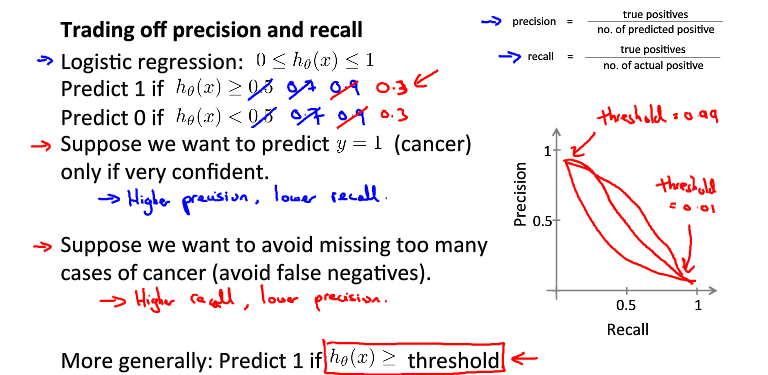
<https://study.163.com/course/courseLearn.htm?courseId=1004570029#/learn/text?lessonId=1053424062&courseId=1004570029>



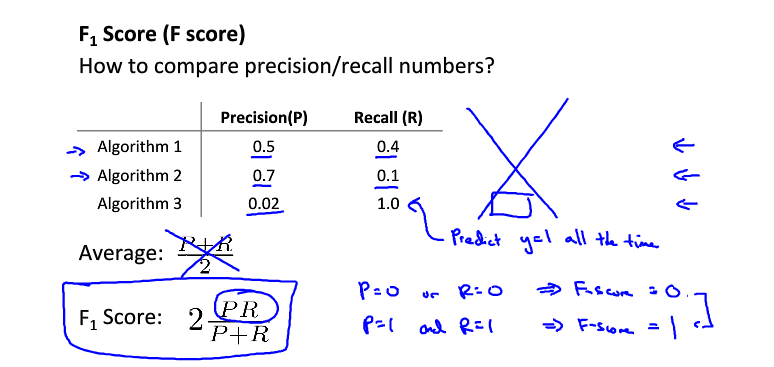
如果测试集只有很少的癌症样本，那么全部预测为无癌症样本则错误率很低，但是并不能代表分类器好！



查准率/查全率（召回率），我们希望这两个值都要高。



设置不同的阀值，得到想要的准确率和召回率。



评估模型时衡量准确率和召回率----

F score.

3、西瓜书：比较检验 <https://zhuanlan.zhihu.com/p/29248751>、

**2019年4月3日**

统计检验（T test）：<https://blog.csdn.net/m0_37777649/article/details/74938120>

<https://blog.csdn.net/bxy5511/article/details/81508262>

<https://blog.csdn.net/myairforce1/article/details/78970203>

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html>

数据预处理参考：D:\pycharm项目\pyradiomics\10.py

<https://blog.csdn.net/weixin_40807247/article/details/82793220>

**2019年4月4日------**

\*特征工程：<http://www.cnblogs.com/jasonfreak/p/5448385.html>

<https://www.zhihu.com/question/28641663/answer/139203996>

**2019年4月7日**

1、随机森林：<https://blog.csdn.net/mao_xiao_feng/article/details/52728164>

<https://blog.csdn.net/colourful_sky/article/details/82082854>

<https://www.cnblogs.com/wj-1314/p/9628303.html>

<https://blog.csdn.net/MG_ApinG/article/details/84872092>

https://blog.csdn.net/u012102306/article/details/52228516

**2019年4月9日**

(Roc)<https://github.com/TryEnlight/Machine-Learning-Projects/blob/master/Random%20Forest%20Tutorial.ipynb>（这个含有ROC/AUC）

画图：<https://blog.csdn.net/ruoyunliufeng/article/details/78076167(Matplotlib)>

**2019年4月11日**

Roc：<https://blog.csdn.net/u013385925/article/details/80385873>

https://blog.csdn.net/qq\_26591517/article/details/80092679（sensitivity:灵敏性；specificity:特异性）

**2019年4月22日**

\*初步代码工程第二次汇总：

我的科研实验代码都在：D:\pycharm项目\pyradiomics

lianxi.py : 这段代码看dicom的文件信息！！！！！！(dicom文件一般会有头文件信息，其文件后缀为dcm)

Lianxi1.py : 获取nii大小！！

Lianxi2.py : 一个dicom序列（series）有很多slice（切片）,这个代码成功将dicom保存nii！！

Lianxi3.py : 批量操作，每个人的术前或术后--进入一个人的术前状态进行dicon合并，合并为nii.gz！

Lainxi4.py--lianxi5.py : 对文件的操作，为了实现数据的批量操作

Lianxi6.py : 批量移动操作，批量移动.gz文件,移动到上一级文件夹

Lainxi7.py--lianxi78.py : 代码虽没有用到，但是是一个思路----主要完成小矩阵向大矩阵赋值！

1.py : 特征提取代码（输入:原图和label，yaml配置文件 ； 输出：按yaml配置的特征值）

2.py : 文件操作，并行跌带两个文件夹下的全部文件

3.py ： 读取csv表格，放入svm训练，此代码还有 GridSearchCV（）网格搜索算法用于调参，数据集划分train\_test\_split（）用法等

4.py : 火狐浏览器2019.3月份关于读取csv表格操作记录等，主要读取csv表格操作读取的csv数据为字符串型，需将其转化为数值型变量

5.py : 判断numpy数组经过tolist()函数变为list，是否与自定义的list等价np.linspace()的使用方法【等差数列的生成】

6.py ： svm成型训练，preprocessing.scale()数据预处理，numpy数组操作

7.py ： numpy数组操作

8.py : 随机种子使用

9.py : T-检验选择特征---from scipy import stats用于进行T检验，pandas保存成csv表格。这是对两种样本的每个特征进行显著性检验，基于原假设，选择出P<0.05的特征

10.py : from sklearn import preprocessing,验证数据预处理---一组特征数据，每一行表示一个样本，每一列表示一个特征

11.py : 此代码用随机森林选择特征,选取对随机森林贡献前10得特征，保存成csv表格

12.py : 尝试可视化随机森林,生成.dot文件

13.py : RF成型训练，调用import evaluate\_model as mpg #自己定义的模块

Batchprocessing.py（D:\pycharm项目\pyradiomics\examples\） : 这个文件为批量处理---提取特征保存为csv数据,根据yaml文件提取特征

evaluate\_model.py (D:\pycharm项目\pyradiomics\) :

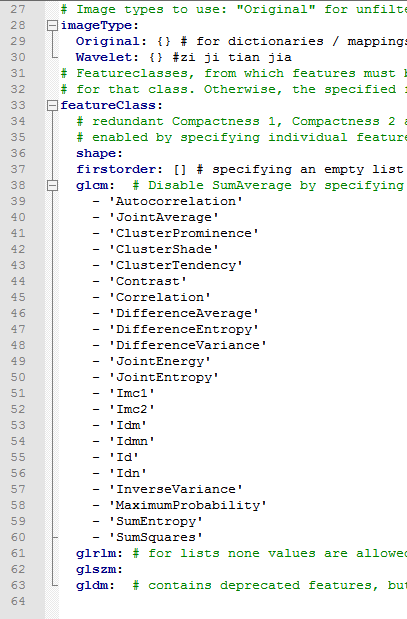
参考：https://www.studyai.cn/modules/classes.html

自定义模块，13.py用到，用于评估模型

输出召回率/查准率/AUC/绘出ROCQ曲线/混淆矩阵（tn, fp, fn, tp）

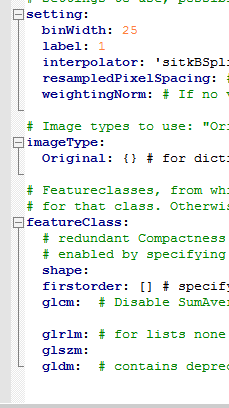
Yaml（用pyradiomics进行特征提取的参数配置文件）:

Params\_test.yaml:（D:\pycharm项目\pyradiomics\）



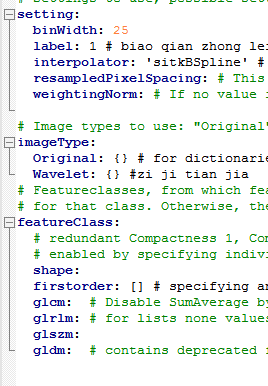
测试用的特征提取参数配置文件

zjParams.yaml:(E:\tls\t2tiqu\zj\_yaml\)



用于对比实验的特征提取参数配置文件

all.yaml(E:\tls\t2tiqu\All\_yaml\):



全特征提取参数配置文件

**2019年4月30日**

特征融合（1）：



AUC代表每个特征对应的ROC曲线下的面积，F代表特征对应的值（Grading of Gliomas by Using Radiomic Features on Multiple Magnetic Resonance Imaging (MRISequences）

特征融合（2）：

即通过lasso进行特征选择，保留系数不为0的特征，然后将保留的特征值与系数相乘取和（CT-based radiomics signature: a potentialbiomarker for preoperative prediction of early recurrence in hepatocellular carcinoma补充；Radiomics Features of Multiparametric MRI as Novel Prognostic Factors in Advanced Nasopharyngeal Carcinoma补充）

**2019年5月5日**

**2019年5月7日**

## **样本不均衡**

<https://blog.csdn.net/u010089444/article/details/70053104>

样本不均衡指的是数据集中的正样本数量与负样本数量的比例失衡。例如，实际应用中，负样本的数量通常远远大于正样本。样本不均衡的危害：造成分类器在多数类精度较高，少数类的分类精度很低，甚至造成分类器失效。解决方案分为以下两种：

1、采样层面

欠采样：通过减少多数类样本来提高少数类的分类性能

随机地去掉一些多数类样本来减小多数类的规模，该方法的缺点是会丢失多数类的一些重要信息，不能够充分利用已有的信息

通过一定规则有选择的去掉对分类作用不大的多数样本（保留与正样本较为接近的负样本）

过抽样：通过改变训练数据的分布来消除或减小数据的不平衡

对少数类样本进行复制，该方法的缺点是可能导致过拟合，因为没有给少数类增加任何新的信息

2、算法层面

改进损失函数的权重，加大少数样本的权值

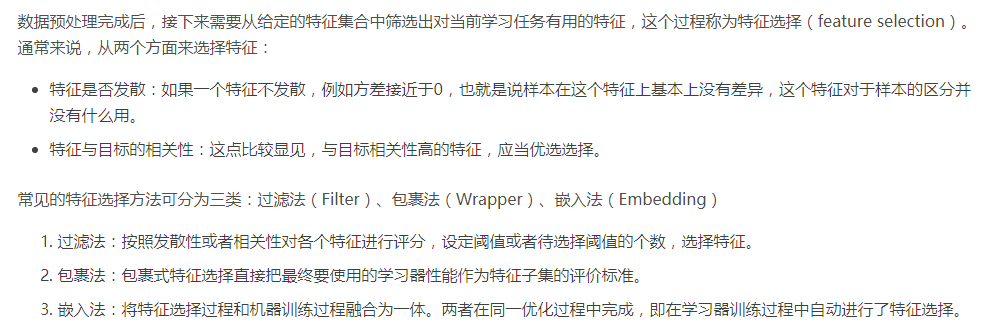
采用集成学习（bagging, boosting）

**2019年5月8日**

**特征选择**

<https://machinelearningmastery.com/an-introduction-to-feature-selection/>

https://machinelearningmastery.com/feature-selection-machine-learning-python/



**2019年5月11日**

阅读pydiomics文档

特征学习

1、First Order Features

Entropy（熵）: <https://blog.csdn.net/marleylee/article/details/78813630>

<https://blog.csdn.net/qq_41205861/article/details/79414596>

<https://blog.csdn.net/mangobar/article/details/80670510>

Percentile（百分位数）：<https://blog.csdn.net/itbuluoge/article/details/29412161>

<https://blog.csdn.net/u011489043/article/details/78056851>

<https://blog.csdn.net/jelek/article/details/7968358>

<https://blog.csdn.net/colorknight/article/details/9531417>

峰度（Kurtosis）和偏度（Skewness）：

<https://blog.csdn.net/xbmatrix/article/details/69360167>

2、Shape Features

3、纹理特征

灰度共生矩阵（glcm）:https://blog.csdn.net/light\_lj/article/details/26098815

<https://blog.csdn.net/xw20084898/article/details/20528997>

https://blog.csdn.net/jiandanjinxin/article/details/51329109

灰度游程矩阵（glrlm）: <https://blog.csdn.net/Beans___Lee/article/details/86579227>

灰度区域大小矩阵（glszm）:

<http://blog.sciencenet.cn/home.php?mod=space&uid=2938101&do=blog&id=976310>

邻域灰度差分矩阵（ngtdm）: <https://blog.csdn.net/Coder_LCB/article/details/90712671>

空间灰度相关矩阵(gldm):

**2019年6月7日**

1. 自定义yaml文件，提取有用的特征

2、量化医生指标（边界模糊；钙化.....）

3、迁移学习，输出深度特征

**2019年6月9日**

数据可视化（画图）

<https://blog.csdn.net/weixin_39506322/article/details/82960476>

<https://blog.csdn.net/yaoxy/article/details/78879152>

z/p值：<https://blog.csdn.net/allenlu2008/article/details/47152263>

**2019年6月11日**

D:\pycharm项目\jiazhuangxian

顺序读取文件夹内容（实现甲状腺图和label并行迭代）

<https://blog.csdn.net/qq_22227123/article/details/79903116>

<https://blog.csdn.net/zj1131190425/article/details/84850018>

<https://blog.csdn.net/Merdy_xi/article/details/78409632>

**2019年6月19日**

实验小问题：

Python中逻辑运算：<https://blog.csdn.net/weixin_40041218/article/details/80868521>

Numpy中squeeze()：<https://blog.csdn.net/tymatlab/article/details/79009618>

<https://blog.csdn.net/tracy_leaf/article/details/79297121>

pandas库pd.to\_csv操作写入数据：

<https://blog.csdn.net/brucewong0516/article/details/79097765>

<https://blog.csdn.net/waple_0820/article/details/70049953>

计算entropy:

<https://blog.csdn.net/shaw820/article/details/72716976>(代码依据)

<https://blog.csdn.net/autoliuweijie/article/details/52244246>

<https://blog.csdn.net/u013617144/article/details/79533868>

<https://blog.csdn.net/qq_24975309/article/details/82026022(用熵进行特征选择)>

<https://blog.csdn.net/qq_32942549/article/details/80019005>

**2019年6月24日**

数组排序：<https://blog.csdn.net/lianshaohua/article/details/80483357>

**2019年7月5日**

插值：<https://blog.csdn.net/u013185349/article/details/84841202>

<https://blog.csdn.net/huozi07/article/details/50538749>

**2019年7月10日**

迁移网络：<https://blog.csdn.net/MiaoB226/article/details/88210189>

<https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html?highlight=transfer>

<https://blog.csdn.net/whut_ldz/article/details/78874977?utm_source=blogxgwz0>

<https://blog.csdn.net/guotong1988/article/details/79739775>

<https://blog.csdn.net/github_36923418/article/details/84567227?tdsourcetag=s_pcqq_aiomsg>

Nii转换为png：<https://blog.csdn.net/weixin_43330946/article/details/89576759>

<https://blog.csdn.net/alxe_made/article/details/80512423>

PIL Image 将numpy的数组转化为PIL Image类型：

<https://blog.csdn.net/yskyskyer123/article/details/80707038>

<https://blog.csdn.net/Hyj6846492/article/details/75041959>

**2019年7月12日**

数据增广：Torchvision transforms 总结

<https://blog.csdn.net/shanglianlm/article/details/85725195>

<https://blog.csdn.net/Hansry/article/details/84071316>

**2019年7月**

做实验

<https://github.com/tony-mtz/CAM/blob/master/CAM.ipynb>

Vgg做2分类解决训练集loss一直为0.693174....

<https://blog.csdn.net/weixin_34343689/article/details/88111552>

Pytorch中自动求导机制错误问题：

<https://zhuanlan.zhihu.com/p/38475183>

关于.size():

若为numpy数组，调用.size不能加括号()即shuzu.size，否则报错

若为tensor,调用时，应写成shuzu.size()，否则并不输出形状，而是输出一段地址

！！！pytorch torchvision.transforms.Normalize()中的mean和std参数---解惑

<https://blog.csdn.net/xys430381_1/article/details/85724668>

<https://blog.csdn.net/york1996/article/details/82711593>

<https://blog.csdn.net/bublebee/article/details/88993467>

ReLU vs PReLU

<https://blog.csdn.net/lanchunhui/article/details/52644823>

<https://blog.csdn.net/guilutian0541/article/details/81119932>

<https://blog.csdn.net/shuzfan/article/details/51345832>

**2019年7月31日**

pytorch: tensor类型的构建与相互转换！！！

<https://blog.csdn.net/JNingWei/article/details/79849600>

pytorch的nn.MSELoss损失函数！！！！！

<https://blog.csdn.net/hao5335156/article/details/81029791>

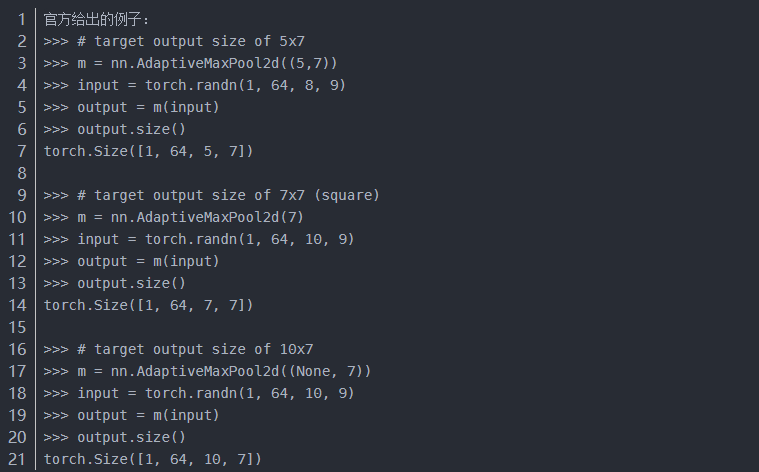
<https://blog.csdn.net/zhangxb35/article/details/72464152>

**2019年8月1日**

Spp实现：（pytorch版）

import torch.nn as nn

nn.AdaptiveMaxPool2d((w, h))



<https://blog.csdn.net/u013382233/article/details/85948695>

<https://blog.csdn.net/xiaosongshine/article/details/89453037>

**2019年8月6日**