1. 混淆矩阵

混淆矩阵是对分类问题的预测结果的总结。使用计数值汇总正确和不正确预测的数量，并按每个类进行细分，这是混淆矩阵的关键所在。混淆矩阵显示了分类模型的在进行预测时会对哪一部分产生混淆。它不仅可以让我们了解分类模型所犯的错误，更重要的是可以了解哪些错误类型正在发生。正是这种对结果的分解克服了仅使用分类准确率所带来的局限性。

在机器学习领域和统计分类问题中，混淆矩阵（英语：confusion matrix）是可视化工具，特别用于监督学习，在无监督学习一般叫做匹配矩阵。矩阵的每一列代表一个类的实例预测，而每一行表示一个实际的类的实例。之所以如此命名，是因为通过这个矩阵可以方便地看出机器是否将两个不同的类混淆了（比如说把一个类错当成了另一个）。

**1.1混淆矩阵组成**



1. .真阳性（True Positive，TP）：样本的真实类别是正例，并且模型预测的结果也是正例，预测正确。
2. .真阴性（True Negative，TN）：样本的真实类别是负例，并且模型将其预测成为负例，预测正确。
3. .假阳性（False Positive，FP）：样本的真实类别是负例，但是模型将其预测成为正例，预测错误。
4. .假阴性（False Negative，FN）：样本的真实类别是正例，但是模型将其预测成为负例，预测错误。

|  |  |  |
| --- | --- | --- |
|  | 公式 | 意义 |
| 准确率(ACC)  Accuracy | Accuracy = | 模型正确分类样本数占总样本数比例（所有类别） |
| 精确率(PPV)  Positive Predictive Value | Precision = | 模型预测的所有positive中，预测正确的比例 |
| 召回率(TPR)  True Positive Rate | Recall = | 所有真实positive中，模型预测正确的positive比例 |
| 特异度(TNR)  True Negative Rate | Specificity = | 所有真实negative中，模型预测正确的negative比例 |

**1.2公式及意义：**

几个指标：

1）正确率（accuracy）

正确率是我们最常见的评价指标，accuracy = （TP+TN）/(P+N)，这个很容易理解，就是被分对的样本数除以所有的样本数，通常来说，正确率越高，分类器越好。

2）错误率（error rate)

错误率则与正确率相反，描述被分类器错分的比例，error rate = (FP+FN)/(P+N)，对某一个实例来说，分对与分错是互斥事件，所以accuracy =1 - error rate。

3）灵敏度（sensitive）

sensitive = TP/P，表示的是所有正例中被分对的比例，衡量了分类器对正例的识别能力。

4）特效度（specificity)

specificity = TN/N，表示的是所有负例中被分对的比例，衡量了分类器对负例的识别能力。

5）精度（precision）

精度是精确性的度量，表示被分为正例的示例中实际为正例的比例，precision=TP/（TP+FP）。

6）召回率（recall）

召回率是覆盖面的度量，度量有多个正例被分为正例，recall=TP/(TP+FN)=TP/P=sensitive，可以看到召回率与灵敏度是一样的。

**1.3其他指标**

1） F1-Score

F1分数（F1-score）是分类问题的一个衡量指标。一些多分类问题的机器学习竞赛，常常将F1-score作为最终测评的方法。它是精确率和召回率的调和平均数，最大为1，最小为0。

对于某个分类，综合了Precision和Recall的一个判断指标，F1-Score的值是从0到1的，1是最好，0是最差。简而言之就是想同时控制recall和precision来评价模型的好坏。

1score=

2）F分数

算出的F1-Score(A)=F1-Score(B),就不太好判断了。为了解决这个问题，人们提出了 Fβ分数。Fβ的物理意义就是将精准率和召回率这两个分值合并为一个分值，在合并的过程中，召回率的权重是精准率的β倍。F1分数认为召回率和精准率同等重要，F2分数认为召回率的重要程度是精准率的2倍，而F0.5分数认为召回率的重要程度是准确率的一半。

score=

3) 错误接受率

FAR (False Acceptance Rate)是错误接受率，也叫误识率，表示错误判定为正例的次数与所有实际为负例的次数的比例。错误判定为正例的次数是FP，所有实际为负例的次数是(FP + TN)。

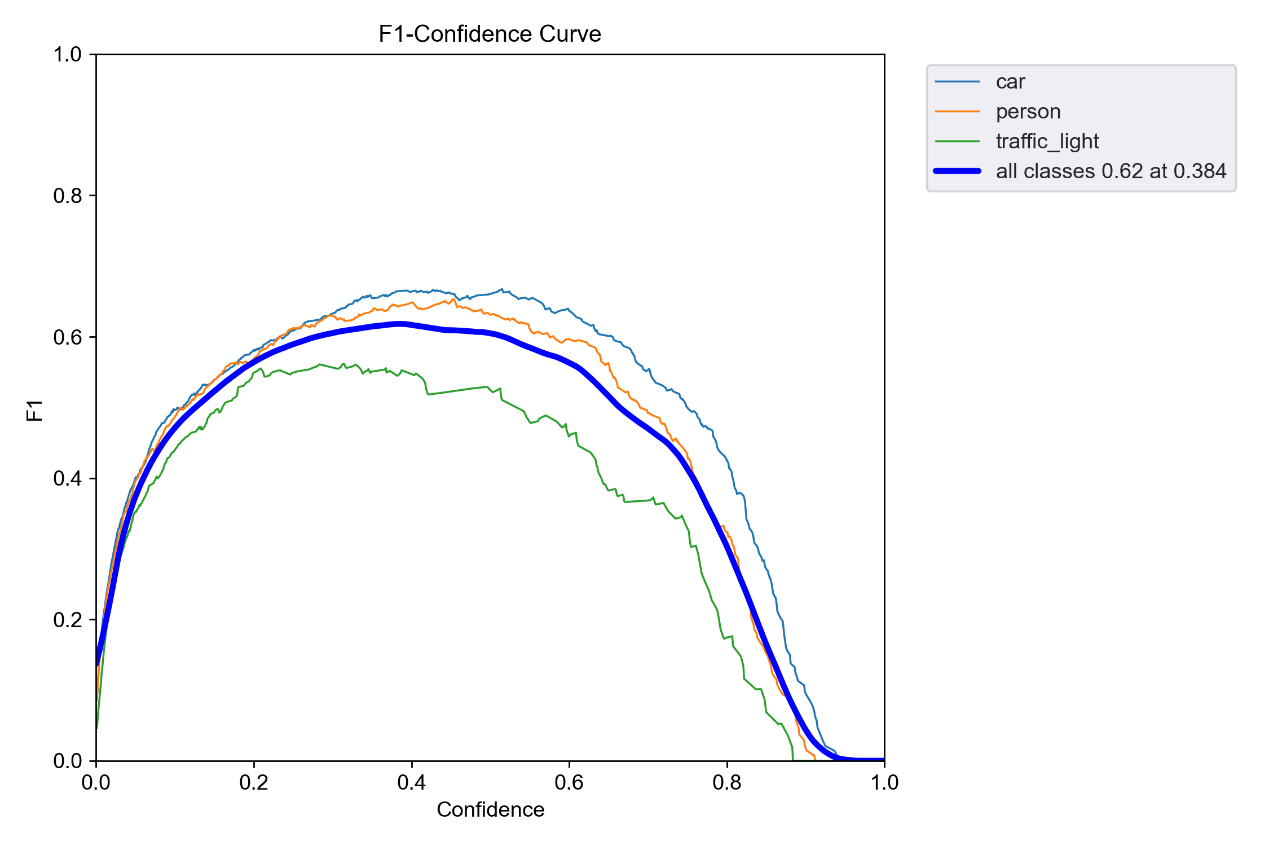
4）错误拒绝率

FRR (False Rejection Rate)是错误拒绝率，也叫拒识率，表示错误判定为负例的次数与所有实际为正例的次数的比例。错误判定为负例的次数是FN，所有实际为正例的次数是(TP + FN)。

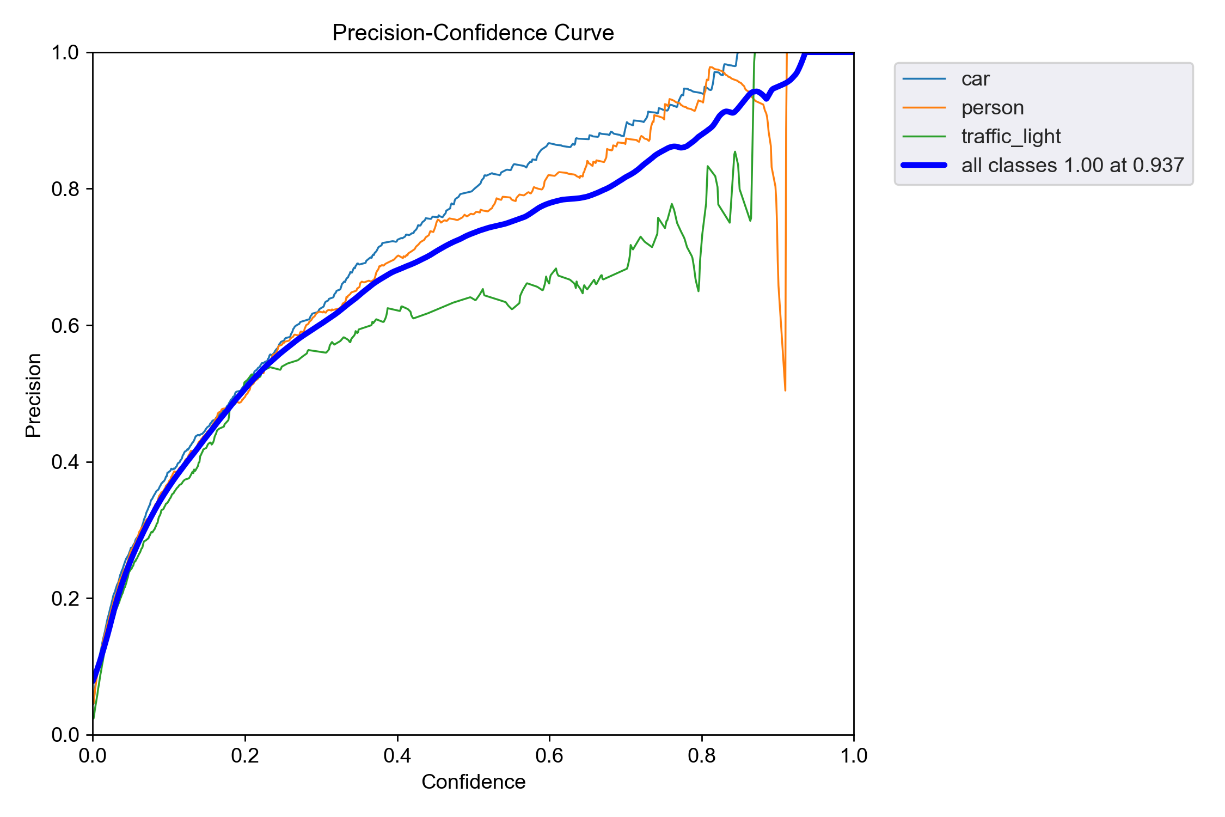
1. 曲线

**2.1 F1分数与置信度曲线**

F1在0-1之间，F1越大，表示性能越好。

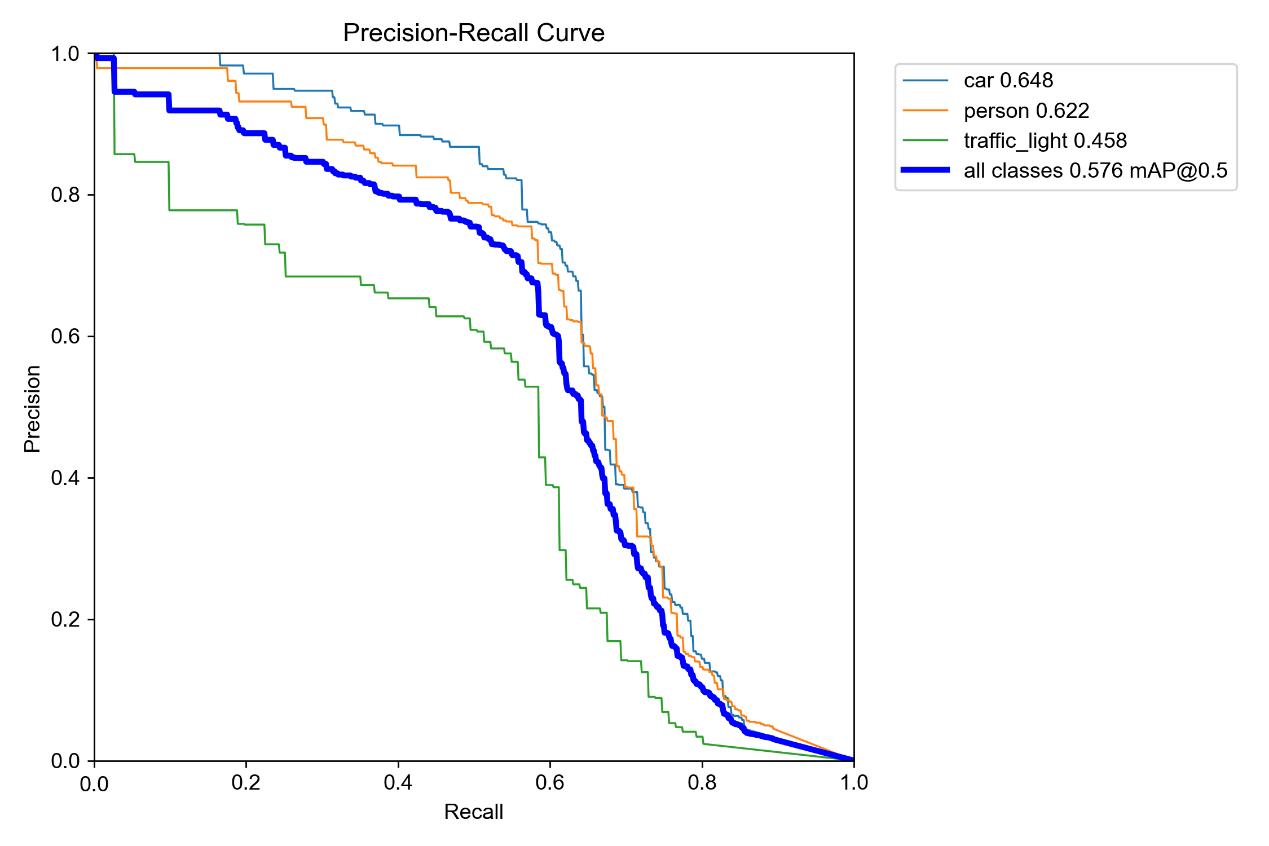


**2.2精度与置信度曲线**

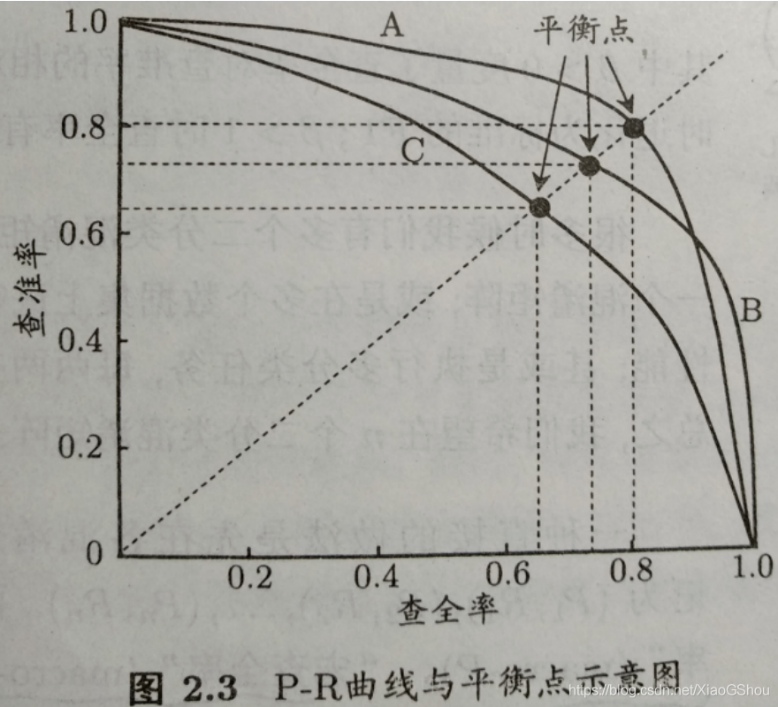


**2.3 PR曲线**

PR曲线中的P代表的是precision（精准率），R代表的是recall（召回率），其代表的是精准率与召回率的关系，一般情况下，将recall设置为横坐标，precision设置为纵坐标。PR曲线下围成的面积即AP，所有类别AP平均值即Map。



如果其中一个学习器的PR曲线A完全包住另一个学习器B的PR曲线，则可断言A的性能优于B。但是A和B发生交叉，那性能该如何判断呢？我们可以根据曲线下方的面积大小来进行比较，但更常用的是平衡点F1。平衡点（BEP）是P=R（**准确率=召回率**）时的取值（斜率为1），F1值越大，我们可以认为该学习器的性能较好。

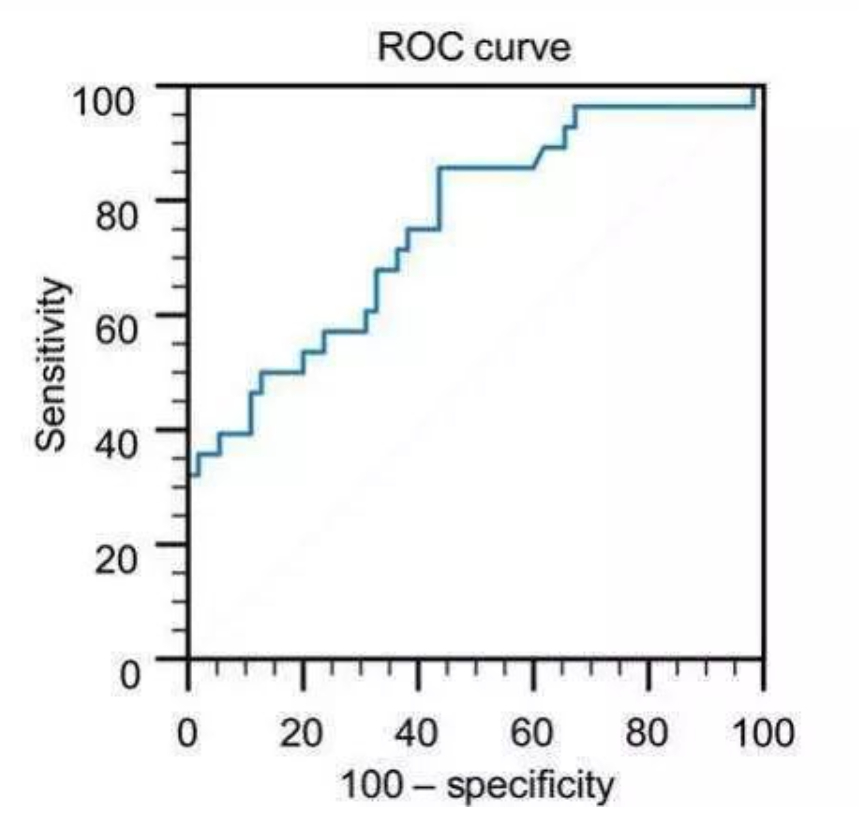


* + 1. **ROC曲线**

ROC 曲线，也称“受试者工作特征曲线”，主要是用于 X 对 Y 的预测准确率情况。最初 ROC 曲线是运用在军事上，现在更多应用在医学领域，判断某种因素对于某种疾病的诊断是否有诊断价值。

ROC 曲线图是反映敏感性与特异性之间关系的曲线。横坐标 X 轴为 1– 特异性，也称为假阳性率（误报率），X 轴越接近零准确率越高；纵坐标 Y 轴称为敏感度，也称为真阳性率（敏感度），Y 轴越大代表准确率越好。

根据曲线位置，把整个图划分成了两部分，曲线下方部分的面积被称为 AUC（Area Under Curve），用来表示预测准确性，AUC 值越高，也就是曲线下方面积越大，说明预测准确率越高。曲线越接近左上角（X 越小，Y 越大），预测准确率越高。



* + 1. **ROC与PR曲线对比**

如何判断一个数据集正负样本是均衡的？

PR和ROC在面对不平衡数据时的表现是不同的。在数据不平衡时，PR曲线是敏感的，随着正负样本比例的变化，PR会发生强烈的变化。而ROC曲线是不敏感的，其曲线能够基本保持不变。

ROC的面对不平衡数据的一致性表明其能够衡量一个模型本身的预测能力，而这个预测能力是与样本正负比例无关的。但是这个不敏感的特性使得其较难以看出一个模型在面临样本比例变化时模型的预测情况。而PRC因为对样本比例敏感，因此能够看出分类器随着样本比例变化的效果，而实际中的数据又是不平衡的，这样有助于了解分类器实际的效果和作用，也能够以此进行模型的改进。

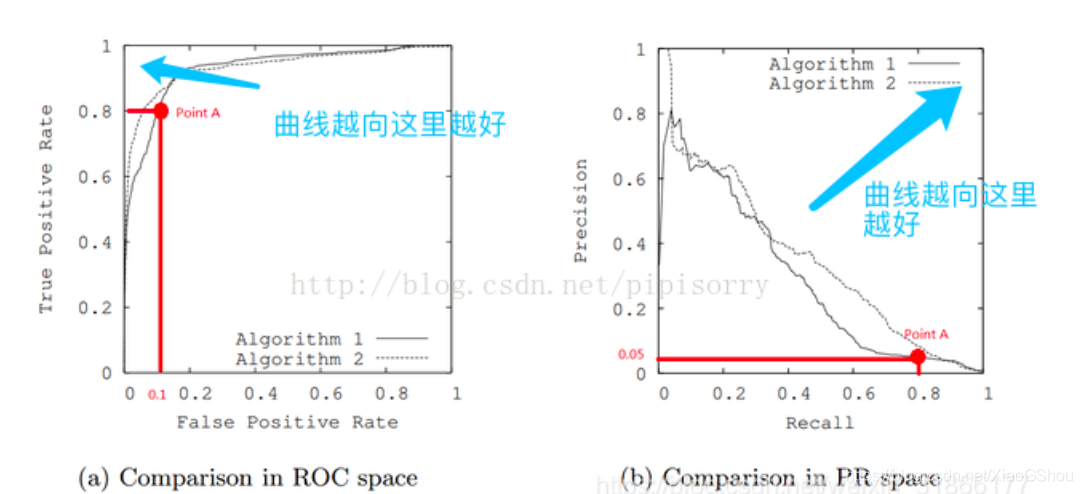
综上，在实际学习中，我们可以使用ROC来判断两个分类器的优良，然后进行分类器的选择，然后可以根据PRC表现出来的结果衡量一个分类器面对不平衡数据进行分类时的能力，从而进行模型的改进和优化。

real world data经常会面临class imbalance问题，即正负样本比例失衡。根据计算公式可以推知，在testing set出现imbalance时ROC曲线能保持不变，而PR则会出现大变化。

PRC相对的优势

当正负样本差距不大的情况下，ROC和PR的趋势是差不多的，但是当负样本很多的时候，两者就截然不同了，ROC效果依然看似很好，但是PR上反映效果一般。解释起来也简单，假设就1个正例，100个负例，那么基本上TPR可能一直维持在100左右，然后突然降到0。这就说明PR曲线在正负样本比例悬殊较大时更能反映分类的性能。

在正负样本分布得极不均匀(highly skewed datasets)的情况下，PRC比ROC能更有效地反应分类器的好坏。



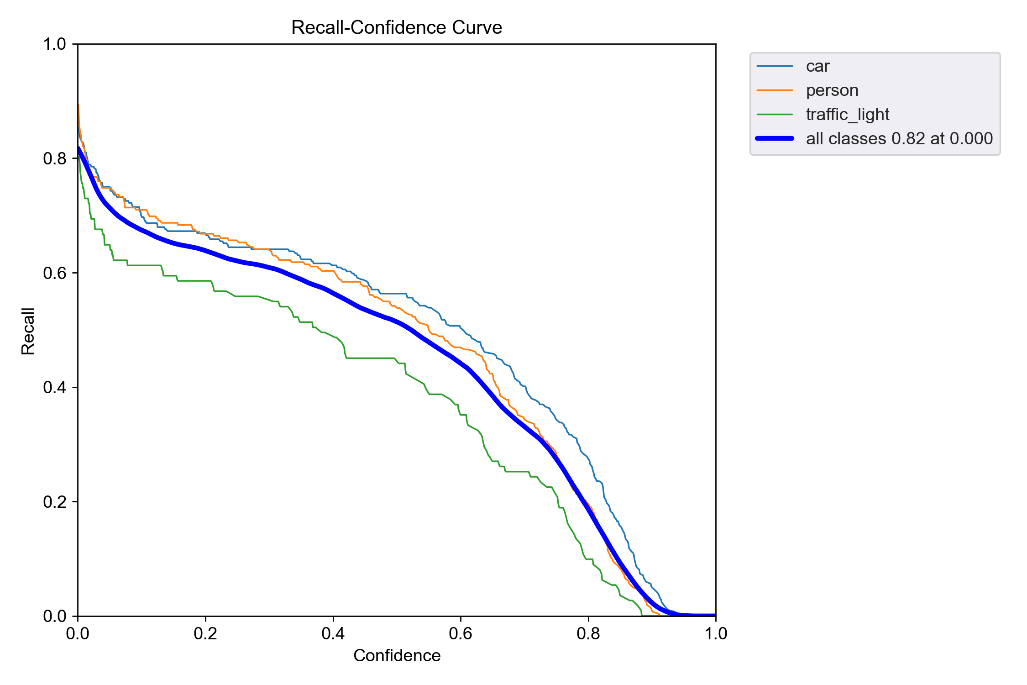
单从图a看，这两个分类器都接近完美(非常接近左上角)。图b对应着相同分类器的PR space。而从图b可以看出，这两个分类器仍有巨大的提升空间。那么原因是什么呢？ 通过看Algorithm1的点 A，可以得出一些结论。首先图a和b中的点A是相同的点，只是在不同的空间里。因为TPR=Recall=TP/(TP+FN)，换言之，真阳性率(TPR)和召回率(Recall)是同一个东西，只是有不同的名字。所以图a中TPR为0.8的点对应着图b中Recall为0.8的点。

假设数据集有100个positive instances。由图a中的点A，可以得到以下结论：TPR=TP/(TP+FN)=TP/actual positives=TP/100=0.8，所以TP=80由图b中的点A，可得：Precision=TP/(TP+FP)=80/(80+FP)=0.05，所以FP=1520再由图a中点A，可得：FPR=FP/(FP+TN)=FP/actual negatives=1520/actual negatives=0.1，所以actual negatives是15200。

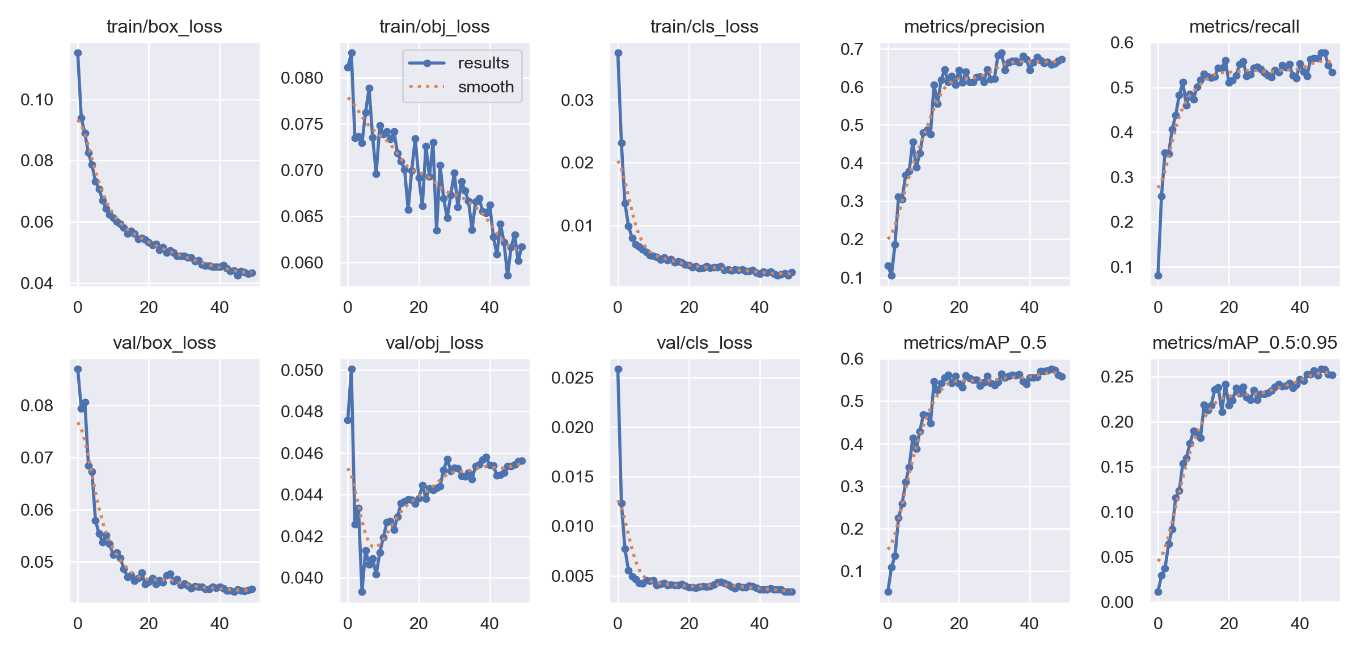
由此，可以得出原数据集中只有100个positive instances，却有15200个negative instances！这就是极不均匀的数据集。直观地说，在点A处，分类器将1600 (1520+80)个instance分为positive，而其中实际上只有80个是真正的positive。 我们凭直觉来看，其实这个分类器并不好。但由于真正negative instances的数量远远大约positive，ROC的结果却“看上去很美”。所以在这种情况下，PRC更能体现本质。

在negative instances的数量远远大于positive instances的data set里， PRC更能有效衡量分类器的好坏。

**2.4 Recall与置信度曲线**



**2.5 Results图**

****

1. Box：YOLO V5使用 GIOU Loss作为bounding box的损失，Box推测为GIoU损失函数均值，越小方框越准；

IOU为交并比， 为同时包含了预测框和真实框的最小框的面积，U为A和B的并集。

2）Objectness：推测为目标检测loss均值，越小目标检测越准；

3）Classification：推测为分类loss均值，越小分类越准；

4）Precision：精度（找对的正类/所有找到的正类）；

5）Recall：真实为positive的准确率，即正样本有多少被找出来了（召回了多少）。

Recall从真实结果角度出发，描述了测试集中的真实正例有多少被二分类器挑选了出来，即真实的正例有多少被该二分类器召回。

6）val BOX: 验证集bounding box损失

7）val Objectness：验证集目标检测loss均值

8）val classification：验证集分类loss均值

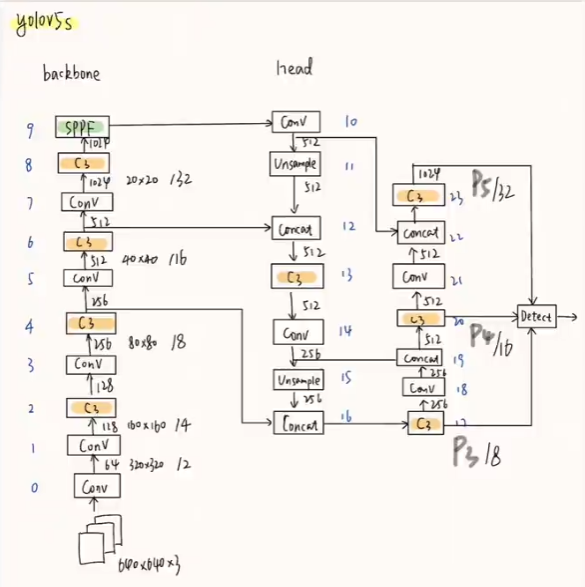
9）mAP@.5：表示阈值大于0.5的平均mAP，一般训练结果主要观察精度和召回率波动情况（波动不是很大则训练效果较好）然后观察mAP@0.5 & mAP@0.5:0.95 评价训练结果。

**10）mAP@.5:.95（mAP@[.5:.95]）：**表示在不同IoU阈值（从0.5到0.95，步长0.05）（0.5、0.55、0.6、0.65、0.7、0.75、0.8、0.85、0.9、0.95）上的平均mAP。

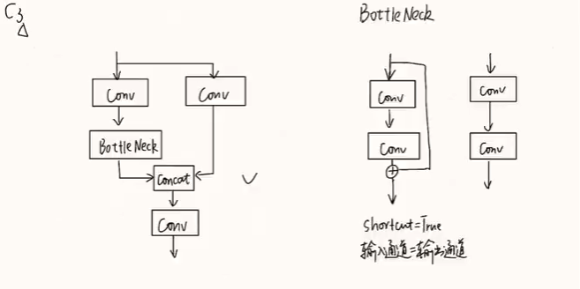
mAP是用Precision和Recall作为两轴作图后围成的面积，m表示平均，@后面的数表示判定iou为正负样本的阈值，@0.5:0.95表示阈值取0.5:0.05:0.95后取均值。

map计算：就是每个recall区间做相应的计算，即每个recall的区间内我们只取这个区间内precision的最大值然后和这个区间的长度做乘积，所以最后体 现出来就是一系列的矩形的面积，还是以上面的那个例子为例，我们一共有recall一共变化了７次，我们就有７个recall区间要做计算，然后实际我们计算的时候人为的要把这个曲线变化成单调递减的，也就是对现有的precision序列要做一些处理。

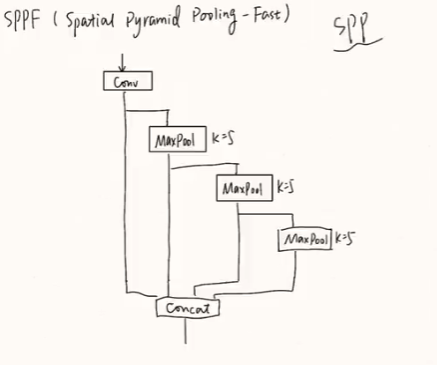
1. 模型讲解



640\*640是尺寸，3是通道，越往上，图片尺寸越小，通道数越多。然后开始先进行一个自上而下的特征融合，再进行自下而上的特征融合。取23、20、17层特征图给detect。



C3是三个卷积，bottleneck里面有跳跃连接



空间金字塔池化快速版