中期作业实验报告

|  |  |
| --- | --- |
| 姓名 | 学号 |
| 查王皓天 | 2018200222 |
| 谢雨涵 | 2019202324 |
| 饶子扬 | 2019202197 |
| 阮雅瑄 | 2018200426 |

[一、 概述 2](#_Toc72015036)

[二、 导入数据 2](#_Toc72015037)

[三、 指标选取 2](#_Toc72015038)

[四、 单分类问题 3](#_Toc72015039)

[（二） PR曲线、ROC曲线、代价曲线 4](#_Toc72015040)

[（三） 五折交叉验证法 6](#_Toc72015041)

[（四） LDA、QDA模型 10](#_Toc72015042)

[五、 多分类问题 12](#_Toc72015043)

[（一） OvO方法 12](#_Toc72015044)

[（二）ECOC三分类 12](#_Toc72015045)

[（三）ECOC五分类 13](#_Toc72015046)

[六、 总结 15](#_Toc72015047)

[七、 思考与讨论 15](#_Toc72015048)

[小组分工 15](#_Toc72015049)

# 概述

在本实验中，我们组运用了logistic回归、LDA、QDA等模型，对Mnist数据集进行了单分类、多分类等问题的探索。**前四题中**，我们组每位组员都选取了十个变量用来提取数据的特征，**总共建立了4个logistic模型**，进行比较，准确率大约都在90%左右，效果较好。**后四题中**，我们选取前四题**4个模型中比较好的两个**，进行后续的多分类问题探讨。在三分类问题中，OVO的准确率达到了75%，而用2个分类器建立的ECOC模型准确率大约为62%。在五分类问题中，第一个模型中，用5个分类器建立的ECOC模型准确率在50%，用10个分类器建立的ECOC模型准确率为58%，用50个分类器建立的模型准确率达到67%。第二个模型表现较差。

# 导入数据

本题中的原始数据为二进制格式，我们借助CSDN上的代码，将其读取至python中的数组格式。读取后，image文件转化为60000\*28\*28的三维数组，label文件转化为60000维向量。

代码参考：[解析train-images-idx3-ubyte与train-labels-idx1-ubyte（mnist数据集）\_醉糊涂仙的博客-CSDN博客](https://blog.csdn.net/u010916338/article/details/85159686)

# 指标选取

在每个图像中，共有28\*28=784个特征变量可供选择，题目要求选取10个特征变量，我们组依次采取了以下四种特征选取方式：

1. **平均灰度从大到小排列，第110到第120的十个格子**

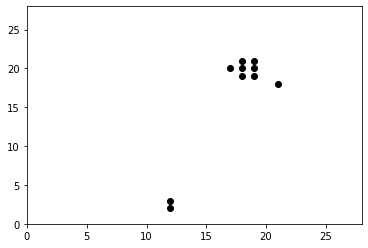
选取理由：所有格子的平均灰度，从大到小排序，依次为3和6都经过的点，3或6都经过的点，3和6都不经过的点。经过尝试，第110到120的十个格子，大约是3或6都经过的点，因此选取其作为特征

1. **标签为3的数据和标签为6对应的灰度矩阵分别加总取平均，二者的均值矩阵作差并取绝对值后，选取该矩阵中最大的前10个格子**

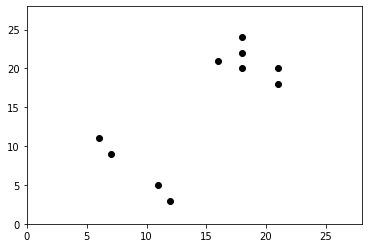
选取理由：由于对分类有用的信息往往是两个类别差距较大的特征，因此在类内取均值，类间作差取最大，得到的前10个灰度特征是3和6间有较大差异的格子，可作为判断依据

1. **灰度值方差最大，且相互之间距离大于2的10个像素点**

选取理由：我们希望找到那些类间差异大的像素点，并用他们进行回归分析，因为他们最可能蕴含了区分两者的特征信息，以这个“差异最大”为出发点，我们打算对所有像素点的灰度值求方差，将其从大到小排序并使用前十个像素点作为回归的依据，得到的结果如图。



观察这样筛选的输出结果后我们发现，所筛选出的像素点聚集较严重，被选中的往往是一个区域内相邻的若干像素点，同一区域内部像素点具有很强的相关性，所包含的信息重复率较高，会降低回归的精度，于是我们对筛选的像素点又增加了一条距离的限制，以尽可能地将更多的类别信息输入到模型之中，最终的筛选规则为：**选取灰度值方差最大，且相互之间距离大于某一阈值的10个像素点作为回归依据。**这样得到的结果如图。



可以看到，加入距离限制之后，变量之间聚集的现象减少，实践证明，加入合适的距离限制条件后（经过尝试大于2时效果最好），模型预测的精确度有显著上升，这里不作展开。

1. **平均灰度每行从大到小排序，取每行最大的灰度值，28个行中随机取10行，定位到10个灰度最大值坐标，即为选择的10个灰度强度值特征**

选取理由：所有格子的平均灰度，在每行内从大到小排序，共有28个每行最大灰度值，考虑到存在3和6重合的像素点，因此随机抽取10个行内最大灰度值坐标，既有代表性又有随机性，选取这10个灰度强度值特征作为变量。

# 单分类问题

1. logistic回归

首先，将3和6的数据分出，然后利用训练集计算出特征，相应的把每个28\*28的矩阵转换成10维向量。至此，数据处理完成（相应代码略，见附件）。

利用sklearn库，在训练集上建立logistic模型，并作用于测试集。结果如下表所示:

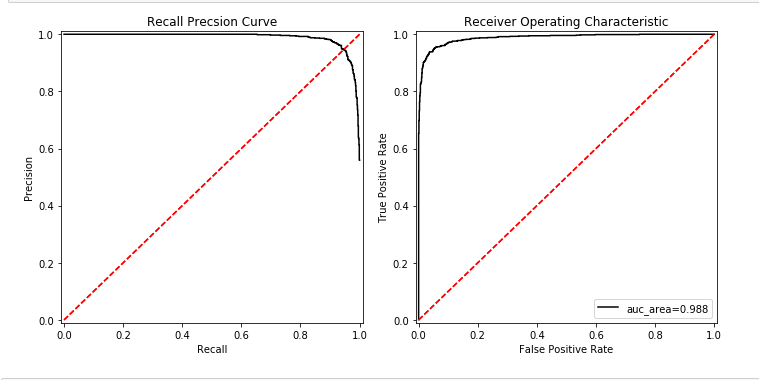
|  |  |
| --- | --- |
|  | 测试集上的准确率 |
| 模型一 | 94.8% |
| 模型二 | 93.7% |
| 模型三 | 92.42% |
| 模型四 | 93.5% |

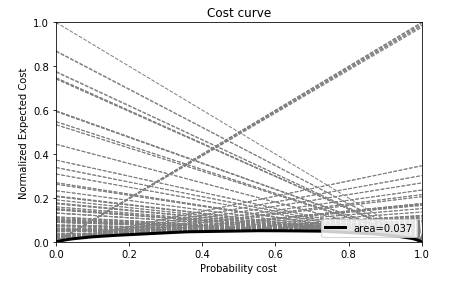
可以看到，二分类准确率均90%在以上

## PR曲线、ROC曲线、代价曲线

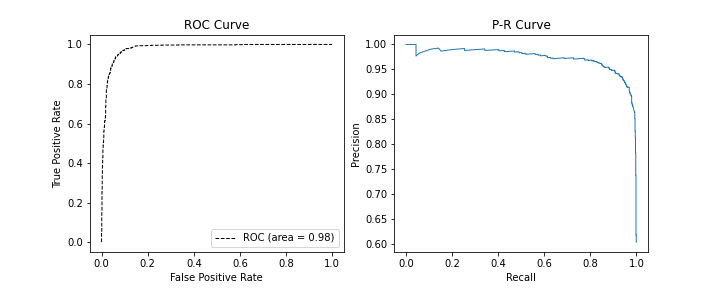
利用sklearn库，绘制PR曲线和ROC曲线。代价曲线利用以往作业中实现的代码（见附件）绘制，若3判错代价是6判错代价的2倍，此种情况下代价曲线和其和横轴围成的面积并不会改变，但是分类器对正类的判断会更加谨慎。图结果如下：

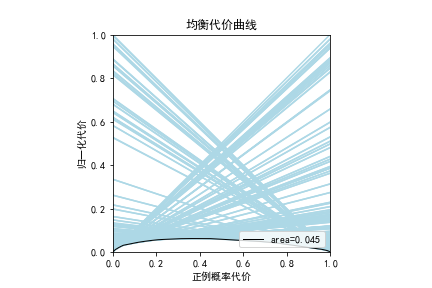
模型一：



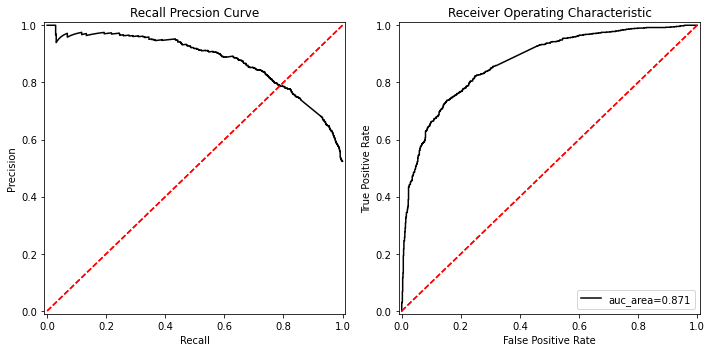


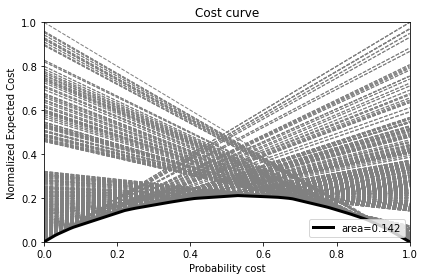
模型二：



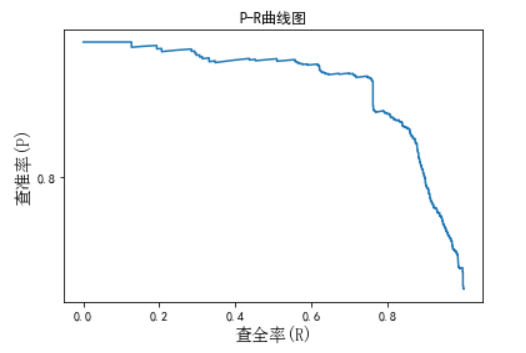
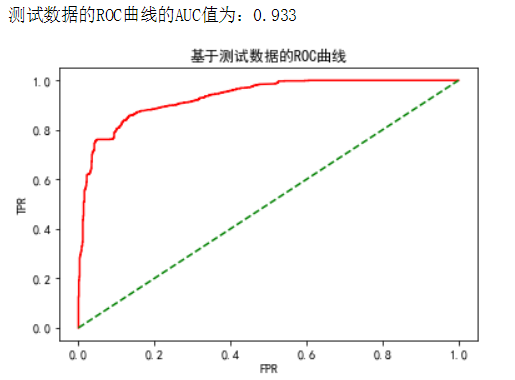


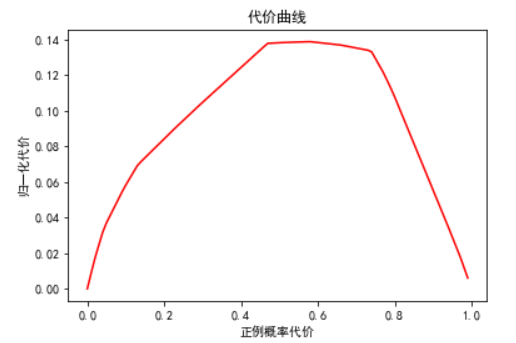
模型三：





模型四：





可以看到，ROC曲线的AUC面积均为0.85以上，而代价曲线下面积均小于0.15，进一步说明分类器效果不错。

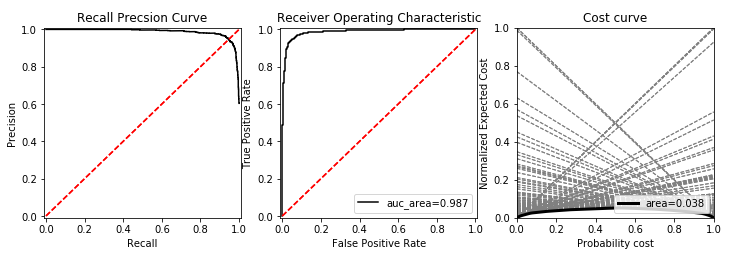
## 五折交叉验证法

模型一：

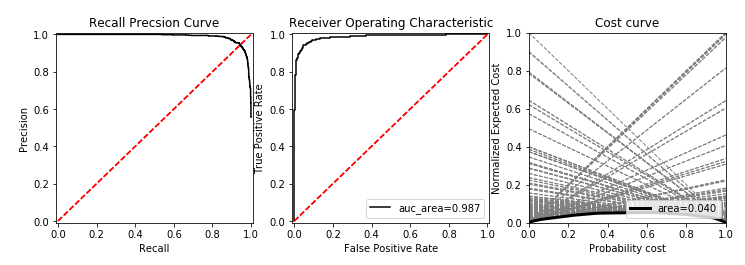
使用五折交叉验证法时，测试集取训练数据中的一部分。利用sklearn库的Kfold函数进行数据划分（代码见附件），同样的模型一和模型二的准确率，roc,pr,cost曲线，（限于篇幅模型34略）：

|  |  |
| --- | --- |
|  | 准确率 |
| FOLD1 | 95% |
| FOLD2 | 95.1% |
| FOLD3 | 96.1% |
| FOLD4 | 94.6% |
| FOLD5 | 94.7% |

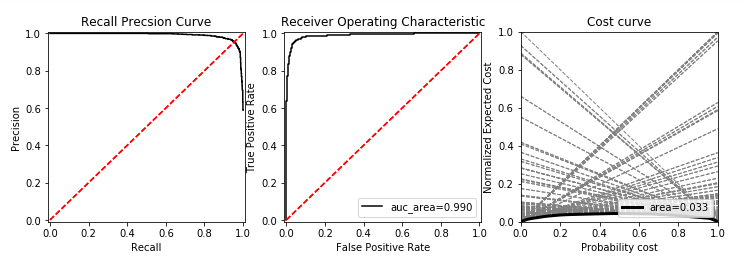
Fold1:



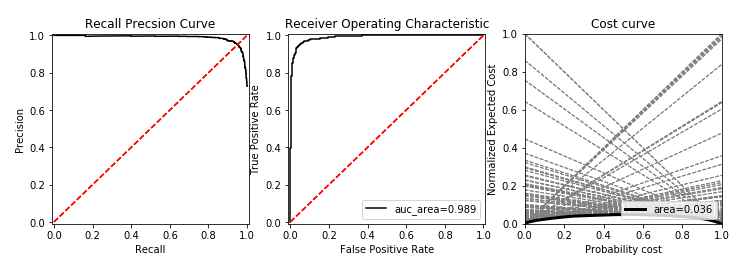
Fold2:



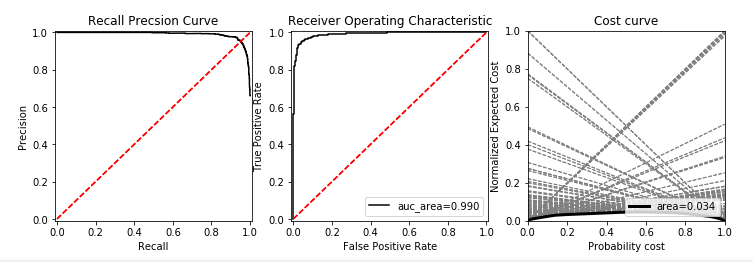
Fold3:



Fold4:



Fold5:

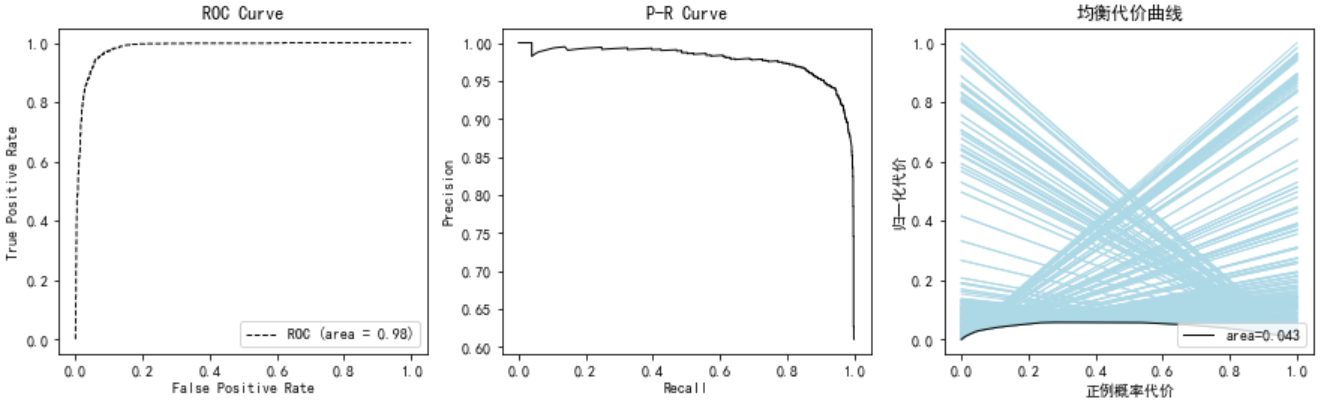


可以看到，每一折的结果相差不多，并且和直接用测试集的结果也相差不大，说明模型是比较稳定的。

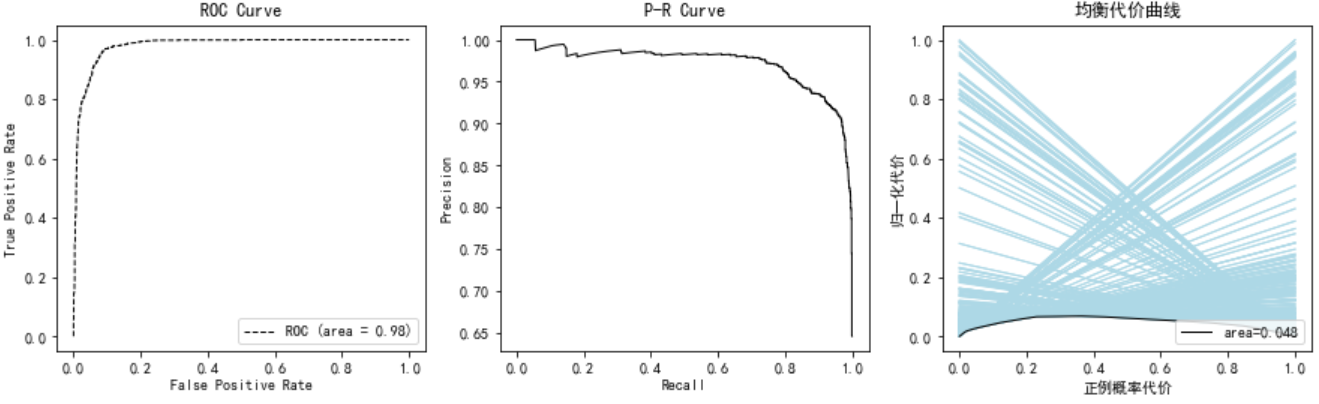
模型二：

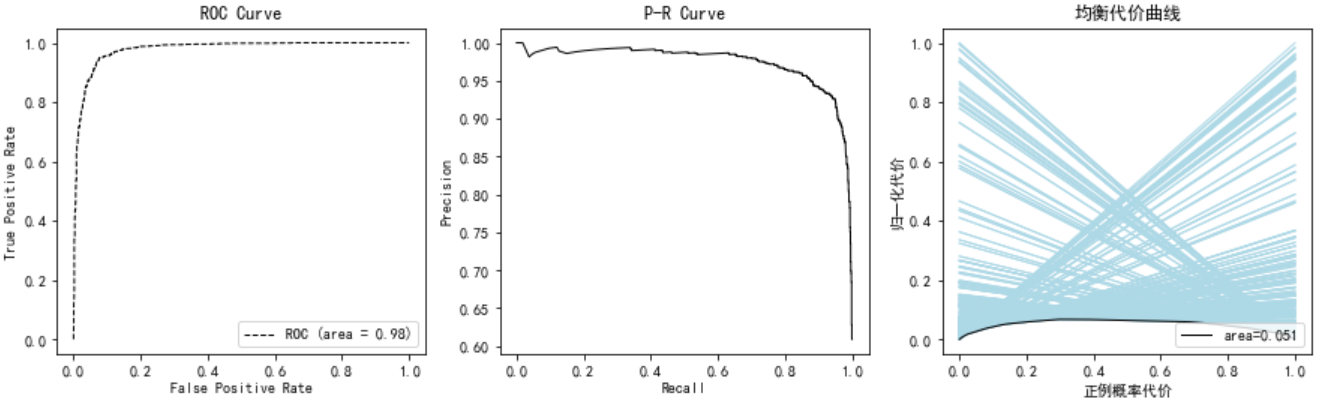
|  |  |
| --- | --- |
|  | 准确率 |
| FOLD1 | 94.12% |
| FOLD2 | 93.30% |
| FOLD3 | 93.33% |
| FOLD4 | 94.65% |
| FOLD5 | 94.18% |

Fold1:

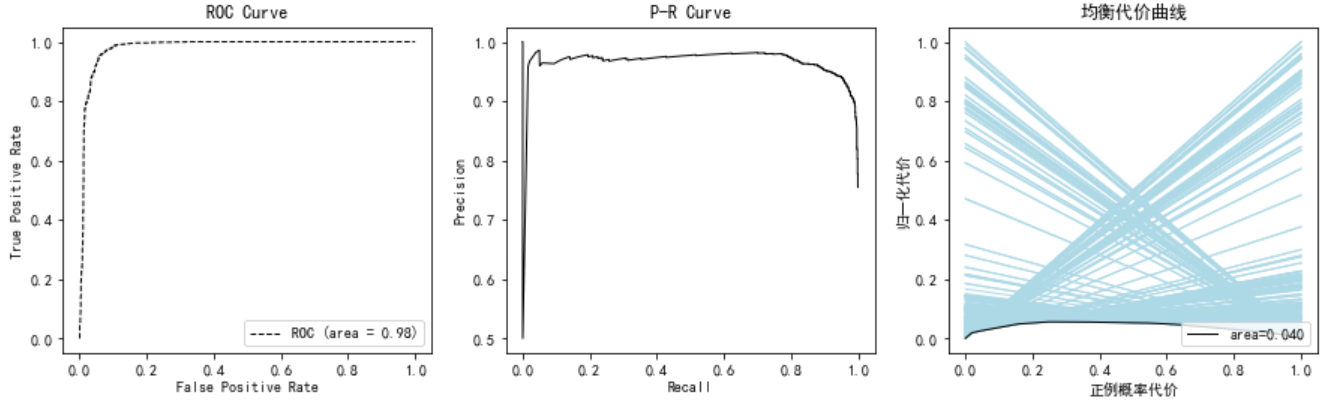


Fold2:

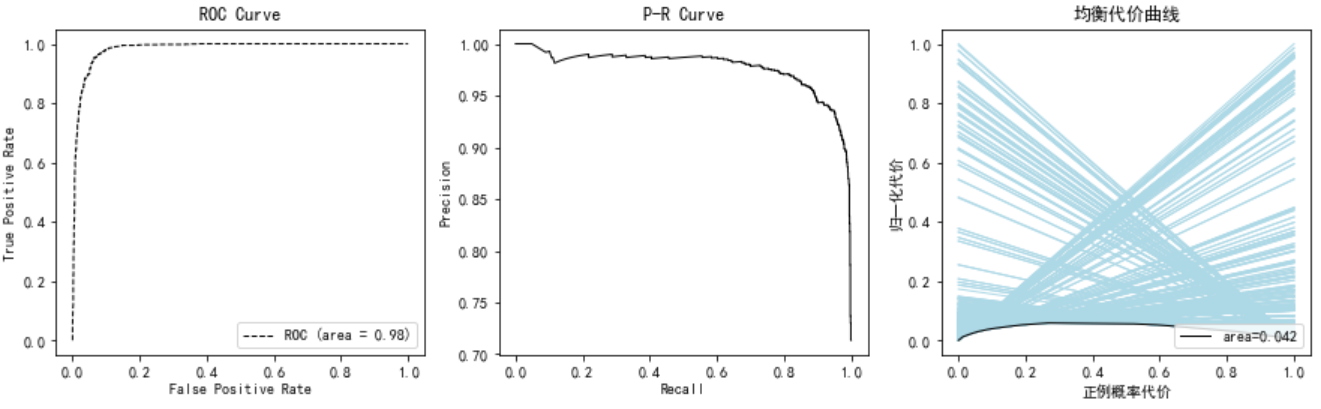


Fold3:

Fold4:



Fold5:



可以看出，所得到的准确率均在90%以上，结果较稳定。

## LDA、QDA模型

*（注：从这题开始，只选取上面四个模型中的两者进行研究，模型一为上面第一个模型，即取平均灰度的模型，模型二为灰度矩阵做差最大十个的模型）*

**模型一：**

该模型下，特征选取为5和9数据中，平均灰度从大到小排列第130-140（调参后的结果）个格子。

将模型一应用到分类5和9的问题中，使用LDA分类方法得到混淆矩阵为：

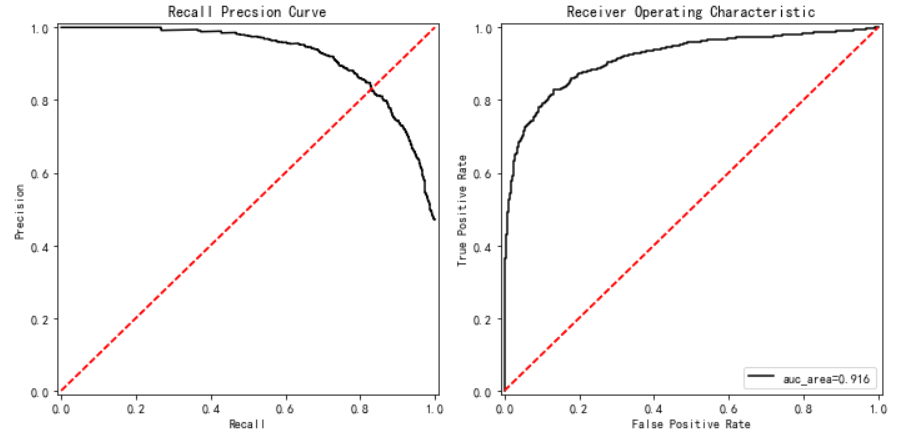
[[875 134]

[154 738]]

得到判别准确率,查准率,查全率,f1-score依次为：

0.84850078905839、0.84633027522936、 0.8273542600897、0.83673469387755

并绘制ROC图和PR图如下：



使用QDA分类方法得到混淆矩阵为：

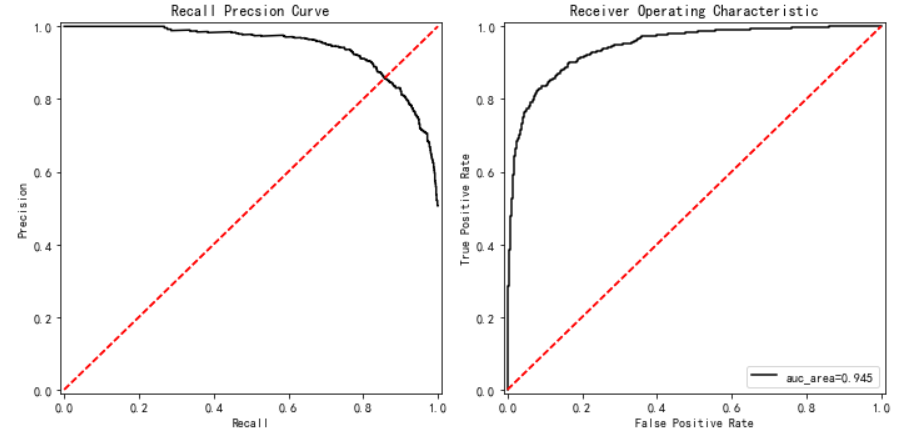
[[884 125]

[128 764]]

得到判别准确率,查准率,查全率,f1-score依次为：

0.8669121514992、0.8593925759280、0.85650224215247、0.8579449747333

并绘制ROC图和PR图如下：



相比之下，QDA分类法的准确率更高。

**模型二：**

该模型下，选取数据中对应的灰度矩阵分别加总取平均并作差取绝对值后最大的前10个格子

将模型一应用到分类5和9的问题中，使用LDA分类方法得到混淆矩阵为：

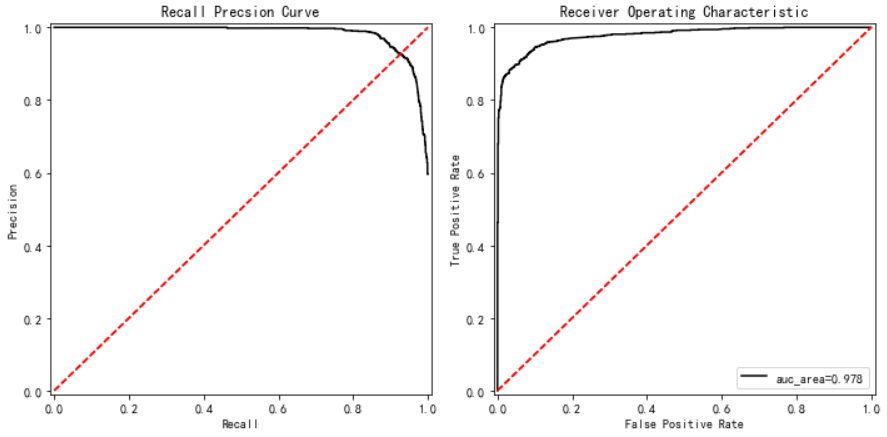
[[821 71]

[75 934]]

得到判别准确率,查准率,查全率,f1-score依次为：

0.9231983166754、0.92935323383085、0.92566897918731、0.9275074478649453

并绘制ROC图和PR图如下：



使用QDA分类方法得到混淆矩阵为：

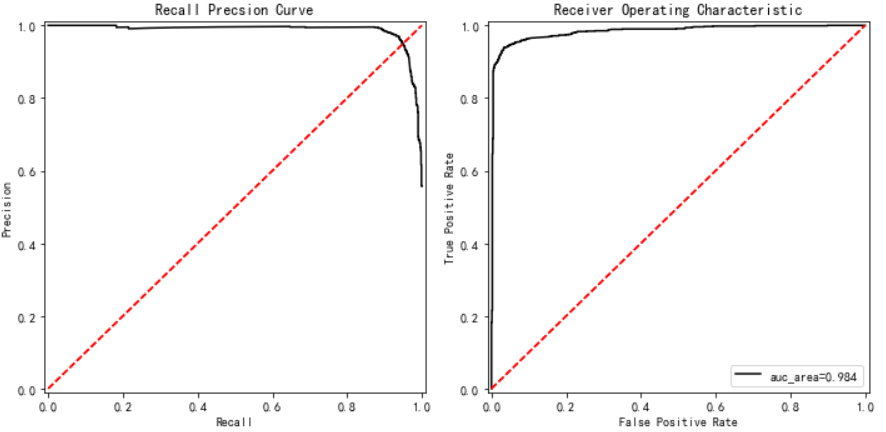
[[861 31]

[62 947]]

得到判别准确率,查准率,查全率,f1-score依次为：

0.95107837980011、0.96830265848671、0.93855302279485、0.9531957725213891

并绘制ROC图和PR图如下：



# 多分类问题

## OvO方法

**模型一：**

该模型下，特征选取为3、5、9三类数据中，平均灰度从大到小排列第110-120个格子。

使用sklearn中封装的OVO进行分类，logistics回归、LDA分类、QDA分类得到的准确率分别是0.716592236、0.71040879、0.764685675

**模型二：**

该模型下，选取5和9数据中对应的灰度矩阵分别加总取平均并作差取绝对值后最大的前10个格子、5和3数据中对应的灰度矩阵分别加总取平均并作差取绝对值后最大的前10个格子、9和3数据中对应的灰度矩阵分别加总取平均并作差取绝对值后最大的前10个格子，三种方法。

使用sklearn中封装的OVO进行分类，logistics回归、LDA分类、QDA分类得到的准确率分别是0.75369288904、0. 7509446925、0. 766059773

## （二）ECOC三分类

根据前文的经验和题目要求，我们在这里基于逻辑回归建立两个分类器：分类器一区分3和5，并分别编码为0和1；分类器二区分3和9并分别编码为0和1。得到3，5，9的编码矩阵如下：

[0, 0]

[1, 0]

[0, 1]

对于回归依据的选择，我们选择了在单分类部分效果最好的指标选取方法（1），即平均灰度从大到小排列，第110到第120的十个格子。

基于此我们分别训练了分类器一和分类器二，并分别对预测集进行分类，得到两列以0和1表示的输出结果，将二者合并，则每一条预测集样本获得了一个形式为[x, y]的编码，分别计算其与3，5，9三者编码的海明距离，取距离最小者归类，最终得到的准确率为62.5%

## （三）ECOC五分类

**模型一：**

该模型下，特征选取为5类数据中，平均灰度从大到小排列第180-190（调参后的结果）个格子。

首先仿照上一题的思路，仍是基于逻辑回归建立了4个二分类器，分别用于判断2/3,3/5,5/8,8/9，四个模型准确率分别为91%,81%,79%,89%。

第二步，建立5\*4的ECOC矩阵，如果某一个分类器把某一类样本分为1的比分为0的多，那么ECOC矩阵相应位置为1，反之为0（代码见附件）。

最后，依次取出测试样本中的每一组特征，依次用4个分类器判断，得到一个1\*4的数组，记录下与ECOC矩阵每一行的欧氏距离，将该样本标记为欧氏距离最小的一类（代码见附件）。最后所有测试数据的准确率为50%。

接下来，尝试增加分类器数量，增加为10个分类器，分别为：

1. 判断2与3
2. 判断3与5
3. 判断5与8
4. 判断8与9
5. 判断9与2
6. 判断23与589
7. 判断35与289
8. 判断58与239
9. 判断89与235
10. 判断29与358

每个分类器单独的准确率均为80%以上，照这样做下来，ECOC最终的准确率提高到58%。

最后，尝试sklearn的函数OutputCodeClassifier进行ECOC，该函数主要是2个参数，第一个参数estimator是选择二分类的模型，第二个参数code\_size，是每个类别要用多少个分类器去分，如果code\_size=1,那么相当于是ovr,所以总共的分类器数量为：类别\*code\_size。



将code\_size设为10，使用总共50个分类器处理上述问题，最终准确率为67%。不过这个函数的问题在于我并不知道他到底用了哪50个分类器。

**模型2：**

运用不同的编码方式和灰度选取方式，其中灰度选取方式共两种：

1. 同第（1）问中模型二方式，对于不同的正负类划分方式，对应的灰度矩阵分别加总取平均，二者的均值矩阵作差并取绝对值后，选取该矩阵中最大的前10个格子。
2. 对于所要分的五类2、3、5、8、9，将其对应灰度矩阵全部加总取均值，取出数值最大的前10个。

以上两种灰度选取方式若单独用在二分类上，第一种准确率可达到90%以上，第二种准确率在60-70%左右。因此方法2划分方式效率低于方法1。

以下为3种所尝试过的方式：

* 选取二元码方式，划分15个分类器（即五个中选一个作为正例，其他作为负例，此有5种，五个中选择两个作为正例，其余作为负例，此有10种（组合数）），得到的编码矩阵如下：

[[1,1,1,1,0,0,0,0,0,0,1,0,0,0,0],

[1,0,0,0,1,1,1,0,0,0,0,1,0,0,0],

[0,1,0,0,1,0,0,1,1,0,0,0,1,0,0],

[0,0,1,0,0,1,0,1,0,1,0,0,0,1,0],

[0,1,0,1,0,0,1,0,1,1,0,0,0,0,1]]

若灰度特征变量选取方式为方法1，利用海明距离求得最终分类类别，得到的准确率仅为36%（代码中未保存，此为最初运用的方式），原因在于此种分类器划分方式得到的各类海明距离有多个重合，说明分类器划分方式有待改进，计算得到的最小值返回的下标也不准确。

* 仍用二元码，改变分类器划分方式，为使各类间每个编码序列与其他良好可分且每一个编码与其余位置编码不相关，构造以下编码矩阵：

[[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],

[0,0,0,0,0,0,0,0,1,1,1,1,1,1,1],

[0,0,0,0,1,1,1,1,0,0,0,0,1,1,1],

[0,0,1,1,0,0,1,1,0,0,1,1,0,0,1],

[0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]]

若灰度选取方式为方式1，最终得到的分类准确率为20.37%。

若灰度选取方式为方式2，则得到的准确率为36.86%。

由于方式1在二分类问题中正确率较低，在多分类中也较低，此处海明距离相对于第一种方法来说重复率有所降低。

* 选取三元码，每次选取两个类分别作为正例和负例，标记为1和-1，其余作为停用类，记为0，构建的编码矩阵为：

[[1,1,1,1,0,0,0,0,0,0],

[-1,0,0,0,1,1,1,0,0,0],

[0,-1,0,0,-1,0,0,1,1,0],

[0,0,-1,0,0,-1,0,-1,0,1],

[0,0,0,-1,0,0,-1,0,-1,-1]]

灰度特征选取采用方法1，最终得到的正确率为38.44%，输出其海明距离后发现距离重复率相对减少但仍存在。

以上尝试的三种方式得到的准确率均较低，思考可能是由于以下原因：

1. 灰度选取方式2在二分类中准确率也相对较低，在五分类中更不理想。
2. 未找到效率高的编码方式，导致每个测试样本计算出的类间海明距离重复率较高，影响最终的类别划分结果，导致判断失误率较高。

# 总结

在本实验中，我们组建了了4个对数字灰度矩阵特征提取方式，并依次完成了基于logistic，lda，qda模型的单分类、多分类问题，绘制ROC、PR曲线、代价曲线，最终在测试集上，单分类达到了90%准确率以上的结果。OvO三分类准确率达到70%+，ECOC三分类的准确率为62%。ECOC五分类下达到了65%的准确率。

# 思考与讨论

**关于特征选取：**

1. 本题我们特征选取的思路都围绕不同数字在不同的格子上，灰度会有所不同，但是忽略了一件事情，如果要认为同一个数字，其灰度矩阵大致是相似的，需要一个假设：**这些数字都在图片的中央**，如果没有这个假设，3在左边和3在右边，其灰度矩阵是完全不同的，所以更好的做法应该是只保留有灰度的部分，这样可以避免上面的问题。这是本实验可以改进的地方。
2. 本实验中，我们特征选取都是单个格子，事实上可能可以进行多个格子的组合，保留更多的信息。

**效率和准确率的矛盾：**

1. 在单分类问题中，我们的准确率达到了95%，是比较好的结果，而在多分类问题上，准确率只有60%，当我尝试使用200个特征来做多分类，即使是5分类，准确率也能有90%以上。如果说，使用784个特征进行分类，效果一定更好，但是那样效率过低。效率和准确性这组矛盾，也是带给我们的思考。如何兼顾这两者？我们认为好的特征可以浓缩信息，在低维解决问题

**多分类问题：**

1. 多分类问题中，我们采取了两种方式，第一种方式，可以事先确定每个数据的特征，而第二种方式，在每次二分类的时候，特征都不同（因为是不同的矩阵相减）。结论发现，第一种方式更有效，也更便捷。我们发现，有的时候从二分类转换到多分类，确实会碰到一些问题（比如本实验中，要用两个矩阵相减，多分类就无法相减），而有的转换，就很自然（比如矩阵相加），这也是一些思考

# 小组分工

我们组的分工如下：

讨论过程所有人都参与，提出自己的想法

做题的分工为：前四题较为简单，每个人建立一种特征选取方式，相应地完成1-4题。5-8题根据题目难度和工作量如下分配：5，6题阮雅瑄，7题饶子扬，8题查王皓天，谢雨涵。

报告写作分工：每个人撰写自己部分，组长最后整合。