



**课 程 报 告**

|  |  |
| --- | --- |
| 课程名称 | 模式识别 |
| 题目名称 | 基于人脸图像的性别与年龄识别 |
| 专业班级 | 2017级自动化创新班 |
| 学号姓名 | 江熠铧 3117000995 |
|  | 蔡瑞涵 3117000985 |
|  | 谭美健 3217000857 |
| 指导教师 | 邢延 |

2020年06月 11日

**目录**

[**1.** **模式识别系统设计** 1](#_Toc42196757)

[**2.** **数据预处理** 1](#_Toc42196758)

[**2.1** **标签转化和删去无效值** 1](#_Toc42196759)

[**2.2** **数据标准化和数据平衡** 1](#_Toc42196760)

[**2.1** **获取原始图像数据** 2](#_Toc42196761)

[**3.** **支持向量机算法** 3](#_Toc42196762)

[**3.1** **支持向量机算法原理与相关概念** 3](#_Toc42196763)

[**3.1.1** **基本原理** 3](#_Toc42196764)

[**3.1.2** **线性可分支持向量机与硬间隔最大化** 3](#_Toc42196765)

[**3.1.3** **线性不可分问题—软间隔支持向量机** 4](#_Toc42196766)

[**3.1.4** **线性不可分问题—非线性支持向量机** 4](#_Toc42196767)

[**3.2** **参数设定与调整** 5](#_Toc42196768)

[**3.2.1** **训练器参数确定与说明** 5](#_Toc42196769)

[**3.3** **实验结果分析与比较** 6](#_Toc42196770)

[**3.3.1** **实验结果** 6](#_Toc42196771)

[**3.3.2** **分析与比较** 8](#_Toc42196772)

[**3.4** **SVM算法总结** 9](#_Toc42196773)

[**4.** **卷积神经网络提取特征+SVM分类** 10](#_Toc42196774)

[**4.1** **卷积神经网络算法原理与相关概念** 10](#_Toc42196775)

[**4.1.1** **基本原理** 10](#_Toc42196776)

[**4.1.2** **卷积层** 10](#_Toc42196777)

[**4.1.3** **池化层** 10](#_Toc42196778)

[**4.1.4** **Dropout层** 11](#_Toc42196779)

[**4.2** **CNN提取特征SVM训练相关理论支撑** 12](#_Toc42196780)

[**4.2.1** **CNN-SVM模型结构** 12](#_Toc42196781)

[**4.2.2** **CNN-SVM性能** 12](#_Toc42196782)

[**4.2.3** **CNN-SVM可行性和复杂度** 13](#_Toc42196783)

[**4.3** **参数设定与调整** 14](#_Toc42196784)

[**4.3.1** **参数调试过程** 14](#_Toc42196785)

[**4.3.2** **模型参数确定** 18](#_Toc42196786)

[**4.4** **实验结果分析与比较** 20](#_Toc42196787)

[**4.4.1** **性别结果** 20](#_Toc42196788)

[**4.4.2** **年龄结果** 22](#_Toc42196789)

[**4.4.3** **结果分析与比较** 24](#_Toc42196790)

[**4.5** **CNN-SVM算法总结** 25](#_Toc42196791)

[5. **SVM算法和CNN-SVM算法比较** 26](#_Toc42196792)

[**5.1** **性能分析** 26](#_Toc42196793)

[**5.2** **复杂程度分析** 26](#_Toc42196794)

[**5.3** **应用的范围** 26](#_Toc42196795)

[References: 28](#_Toc42196796)

1. **模式识别系统设计**

内容要求：

说明识别的目标和类别，例如：识别目标：性别，类别：男和女两类、

数据来源及数据特点（数据量、特征维度、类别是否均衡、数据量是否足够等）

采用的开发工具（编程语言、集成开发环境等、要求用Python）

采用的模式识别方法（注意要与数据特点相匹配）

是否做特征提取与选取，若是，采用什么算法（经典、改进）

采用的分类算法（含经典算法、智能算法/改进算法）

分类器性能评价方法（K折交叉验证，分类准确率/错分率/AUC,是否考虑算法的时间复杂度等）

1. **数据预处理**
2. **标签转化和删去无效值**
3. **数据标准化和数据平衡**

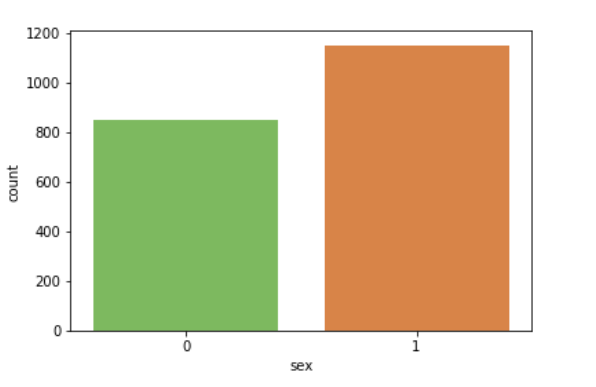


图2-3 数据性别分布

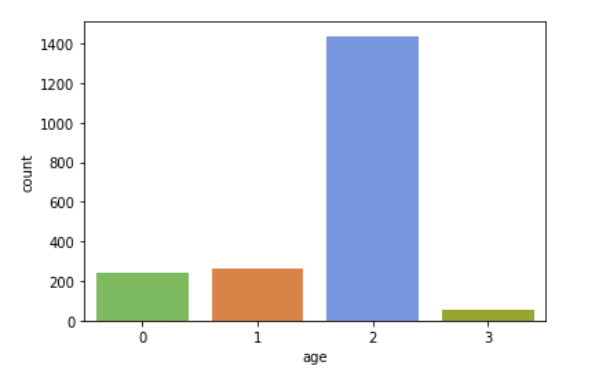


图2-3 数据年龄分布

1. **获取原始图像数据**
2. **支持向量机算法**
3. **支持向量机算法原理与相关概念**

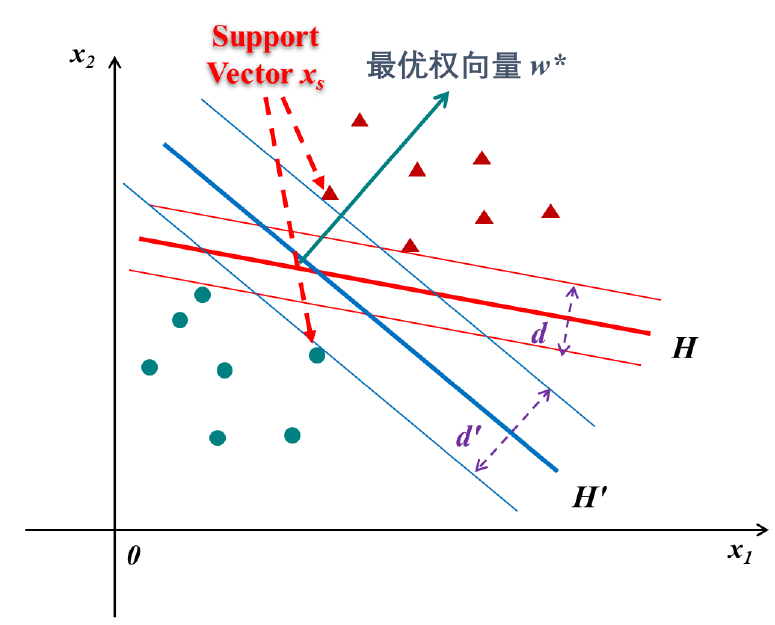


图3-1 SVM基本原理

1. **基本原理**

支持向量机是一种二分类模型，基本模型（如上图）是定义在特征空间上的间隔最大的线性分类器。因此，SVM的学习策略就是**间隔最大化**，学习算法就是求解**凸二次规划**的最优化算法。

1. **线性可分支持向量机与硬间隔最大化**
2. **线性可分支持向量机**：给定线性可分训练数据集，通过间隔最大化或等价地求解相应的凸二次规划问题学习得到的**分离超平面**为

以及相应的**分类决策函数**

称为线性可分支持向量机。

1. **函数间隔和几何间隔**

函数间隔即训练集T中所有样本点距离超平面的**距离**

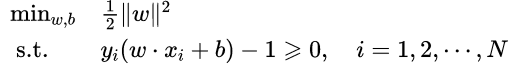
几何间隔，在对于超平面发现量加上某些约束时，如规范化后，，使得间隔是确定的，这时的函数间隔就是几何间隔，常表示为：

1. **最大间隔法**

输入：线性可分训练数据集

输出：最大间隔分离超平面和分类决策函数

构造并求解约束最优化问题：



求解最优

得到超平面和决策函数：

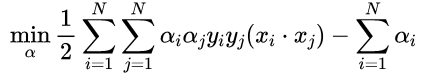
由于最大间隔分离超平面存在**唯一性**，因而能够找到使得两类点完正确分开的分离超平面。

1. **拉格朗日乘子法**

对优化问题中，每一个不等式约束引入**拉格朗日乘子**得到拉格朗日函数及拉格朗日乘子向量。进而根据其对偶性，**求解极大极小问题**：



得到



因而可以利用上述约束函数求解最优值，进而求得分离平面和决策函数

1. **线性不可分问题—软间隔支持向量机**

对于线性不可分训练数据，显然上述的线性可分支持向量机是不适用的。为了解决这一类情况，将硬间隔最大化改为**软间隔最大化**。

线性不可分意味着某些样本点不满足函数间隔大于等于1的约束条件，修改约束条件引入松弛变量，得到**约束条件**为：

**目标函数**变为：(C>0惩罚参数)

1. **线性不可分问题—非线性支持向量机**

对于解线性分类问题，线性分类支持向量机有很好的性能，但是实际情况中许多分类问题都是非线性的。这时可以利用核技巧，实现低维空间到高维空间的映射计算，进而得到非线性支持向量机。

**核函数**包括：

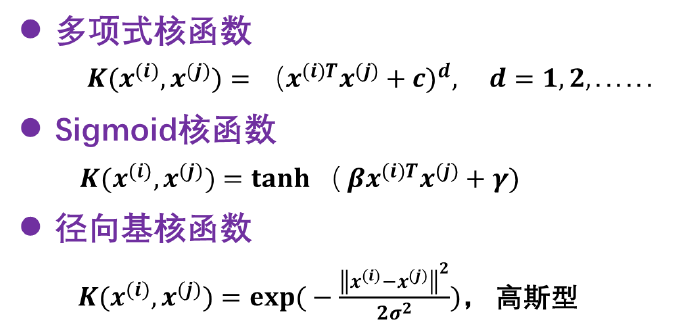


图3-2 SVM核函数

1. **参数设定与调整**
2. **训练器参数确定与说明**

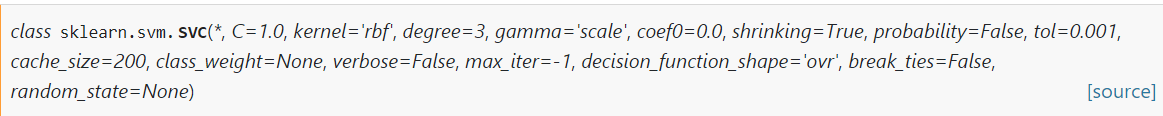
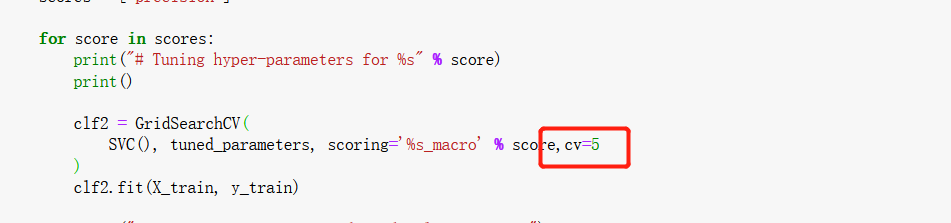


图3-3 SVC函数参数

|  |  |
| --- | --- |
| 参数 | 说明及参数的选择 |
| C | 惩罚因子，采用L2正则化 |
| 选用0.1，1，10三个不同量级的参数去训练，以观察不同正则化程度对分类器的影响 |
| kernel | 指定算法的内核类型，可选择‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} |
| 结合本次的数据要求，选择linear’, ‘rbf’进行训练 |
| Probability | 是否启用概率估计，这个在内部会采用5折交叉验证 |
| 为更好的寻求一个最优分类器，此次选择False和True |
| verbose | 启用详细输出 |
| 选择true，观察训练过程的输出 |
| class\_weight | 调整类别权重 |
| 可选用，但因为本次数据已经经过SMOTE算法进行平衡这里就保持默认值 |
| 其他未调整的参数保持默认值 | |

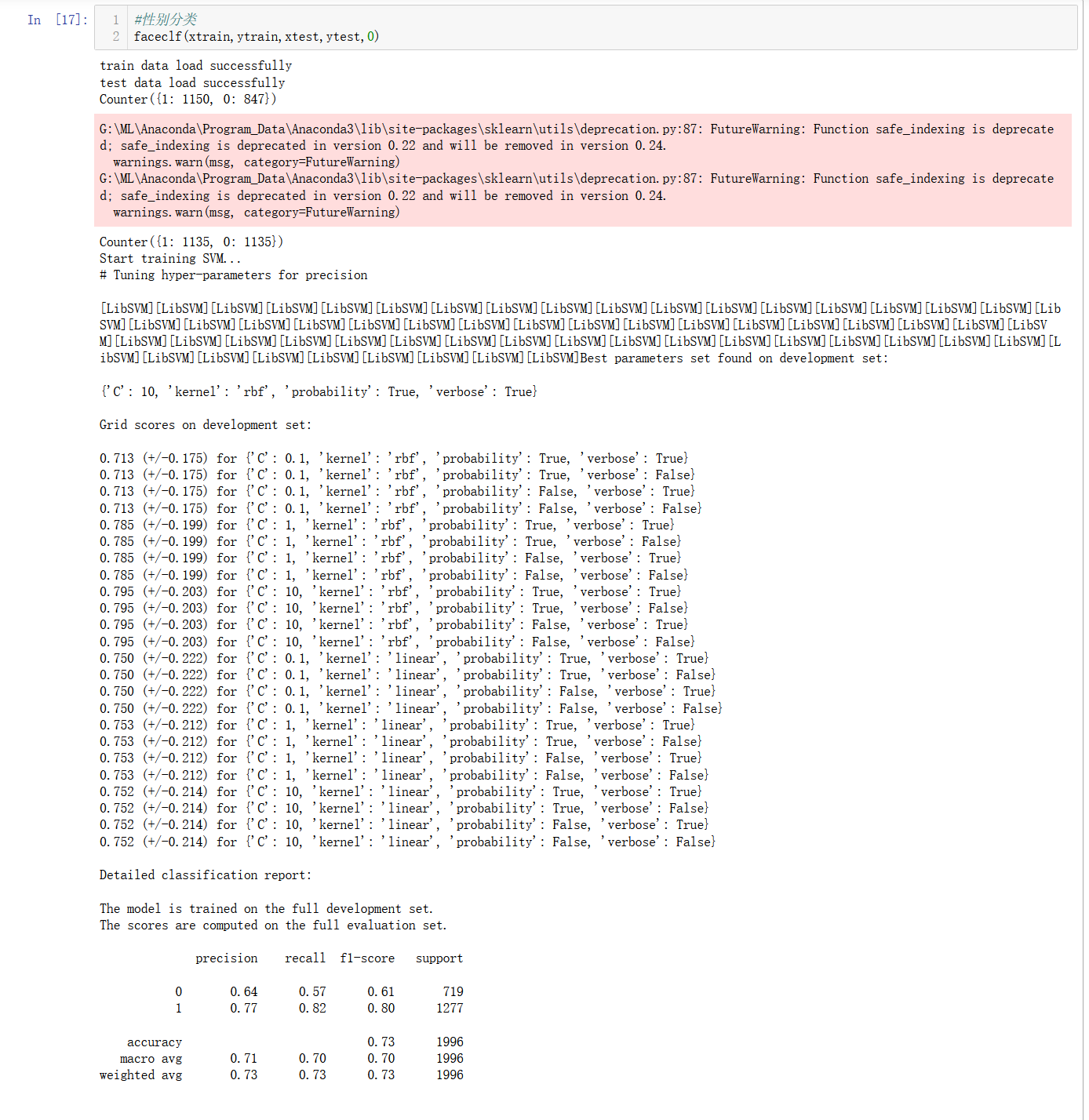
表3-1 参数选择与说明

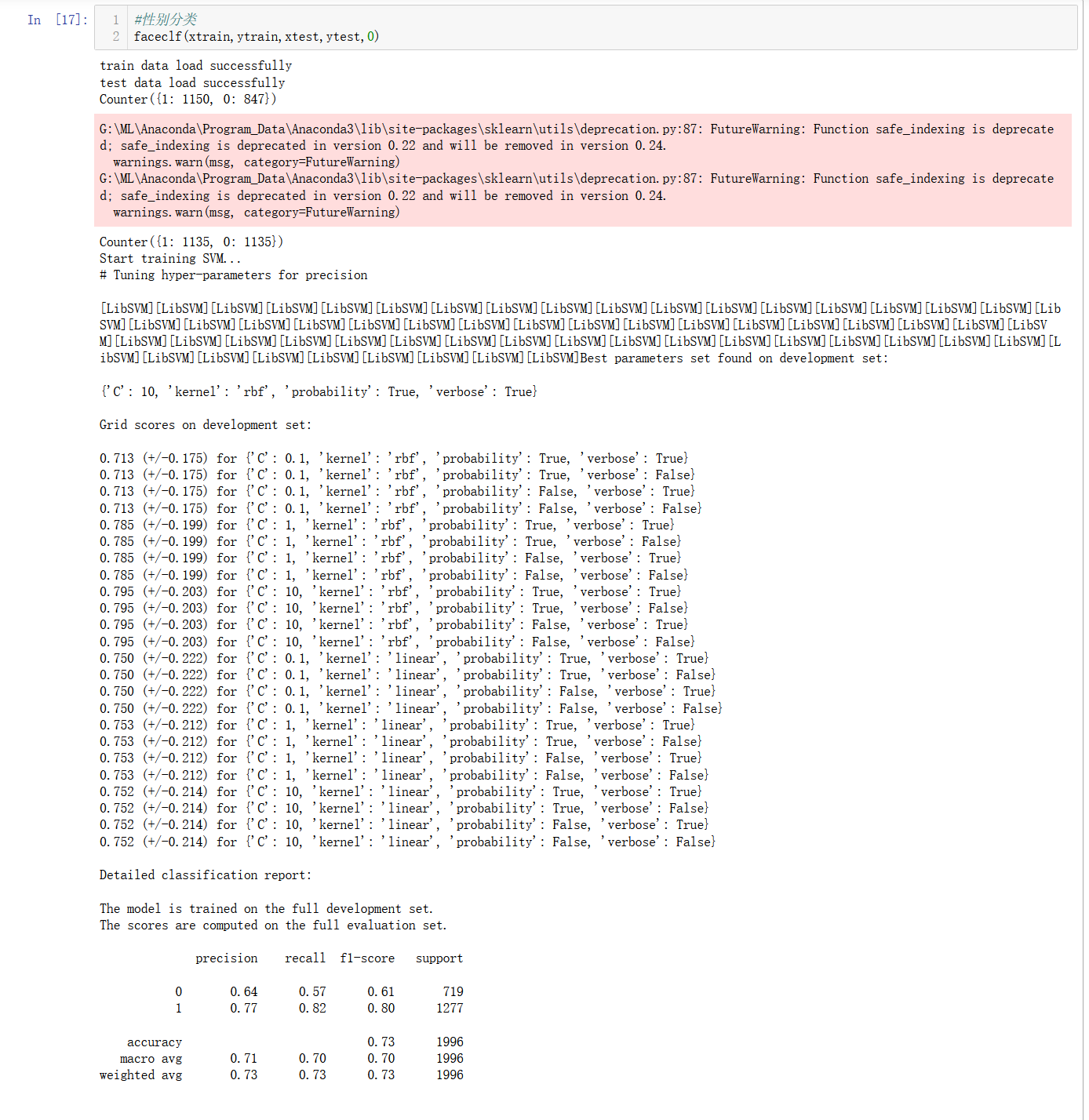


如图所示，本次实验采用5折交叉验证。

1. **实验结果分析与比较**
2. **实验结果**

* **性别结果**：





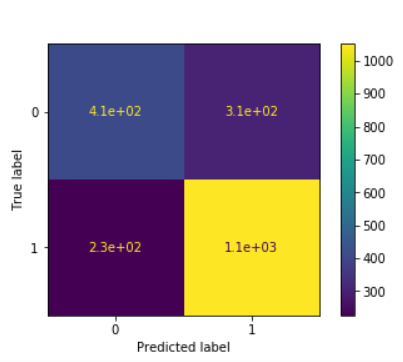


图3-4 Confusion matrix-sex

* **年龄结果**



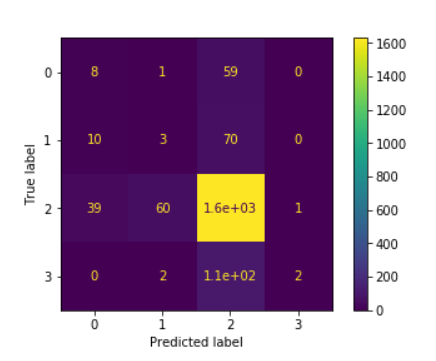


图3-5 Confusion matrix-age

1. **分析与比较**

从两次的寻找最优参数的过程中，可以发现实际上'probability'这个参数的变化对于训练集的准确率没有产生太大的影响，主要影响到准确率的是惩罚因子C和核函数kernel的选择。

从kernel的调节来看，对于两组数据样本的进行训练，当kernel=’linear’时，改变惩罚因子C并没有使得分类器对于训练样本有比较大程度的性能变化。反观，当kernel=’linear’时，改变惩罚因子C，可以明显的看到分类器的性能有明显的变化。

从惩罚因子C来看，有着越大准确率越高的趋势，也符合正则化减小过拟合的理论依据。但考虑到正则化的性质，当太大的时候可能会导致出现欠拟合，此次实验采用了三个量级的参数较为合理。

从两个混淆矩阵的结果来看，可以很明显的看到每一个类别在测试集中的占比有很大的不同，并且结合我们原本所有的类别数据训练时也同样如此，因此分类器性能在很大的程度上受数据的影响。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prediction | | | | |
|  | Precision | Recall | F1-score | support |
| 0-female | 0.64 | 0.57 | 0.61 | 719 |
| 1-male | 0.77 | 0.82 | 0.80 | 1277 |
| accuracy |  |  | 0.73 | 1996 |
| Marco avg | 0.71 | 0.71 | 0.70 | 1996 |
| Best parameters | | | | |
| {'C': 10, 'kernel': 'rbf', 'probability': True, 'verbose': True} | | | | 0.795 (+/-0.203) |

表3-2 性别分类结果

从上表数据我们可以看到分类器对于测试集样本的准确率虽相比于训练时有一定程度的降低，但也比较符合实际情况（训练集：79% 测试集：73%）。从具体的数据来看，两个类别的查准率，查全率和F1值存在一定的差距。但是结合我们实际用到的数据来看，原来的数据样本的分布存在一定的差异（0:847,1:1150），在采用SMOTE算法对数据进行平衡后的数据在一定程度上和实际数据还是有差异的，所以导致了”female”类别测试准确度要相对低一些。整体来说虽然准确度并不是很高，但是也比较符合实际情况。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prediction | | | | |
|  | Precision | Recall | F1-score | support |
| 0-child | 0.14 | 0.12 | 0.13 | 68 |
| 1-teen | 0.05 | 0.04 | 0.04 | 83 |
| 2-adult | 0.87 | 0.94 | 0.91 | 1730 |
| 3-senior | 0.67 | 0.02 | 0.03 | 115 |
| accuracy |  |  | 0.82 | 1996 |
| Marco avg | 0.43 | 0.28 | 0.28 | 1996 |
| weighted avg | 0.82 | 0.82 | 0.79 | 1996 |
| Best parameters | | | | |
| {'C': 10, 'kernel': 'rbf', 'probability': True, 'verbose': True} | | | | 0.976 (+/-0.071) |

表3-3 年龄分类结果

结合训练结果来看，我们的训练在一定程度上存在过拟合的情况，因为训练集训练准确率可以达到97.6%，但是测试数据才82%，相差了15%的精度。同时，可以清楚的看到，除了adult这一个类别的查准率，查全率和F1值相对比较高之外，其他的类别分类效果并不是特别好。从F1值和support可以明显看出这个分类器预测图片年龄可能会在大多数情况下直接选择adult这一个类别，那么由于本身测试集和训练集中包含adult类别远远超过其他类别数。由于每个类别相差太远，经过SMOTE算法也不能很好的对于其他类别生成与实际完全一致的数据，那么也就导致存在过拟合的情况出现。

1. **SVM算法总结**

在这一次训练中，我直接采用了已经进行PCA降维之后的特征数据对性别和年龄两个类别进行分类。实验所用到数据是通过标准化和数据平衡的。训练分离器算法用到的是支持向量机。整体的测试集预测效果综合来看是对于性别的预测相对要好一些，因为本身由于两个类别的样本数目有差异，但没有像年龄存在相差1000左右类别的数据。其实两个样本训练的分类器的性能都比较差，性别分类的准确率不高，年龄分类的存在“谁多猜谁“的现象F1值在少数样本偏低。

本次实验总的来看，其实受到数据分布影响比较大，并且由于一些样本数据较少而导致了一定的过拟合现象、准确率不高的情况。虽然看上去数值比较“看得过去“，但其实内核的分类器性能并不是特别优秀，还有很多可以改进的地方。

一方面，可以采用数据增强的手段，通过对原始数据进行旋转、缩放等操作增加样本数不足的类别。

另一方面，也可以对原始数据进行降维操作，选择更多维度的数据，可以更好的反映图像的特征。

1. **卷积神经网络提取特征+SVM分类**
2. **卷积神经网络算法原理与相关概念**
3. **基本原理**

卷积神经网络与普通神经网络非常相似，它们都由具有可学习的权重和偏置常量(biases)的神经元组成。每个神经元都接收一些输入，并做一些点积计算，输出是每个分类的分数，普通神经网络里的一些计算技巧到这里依旧适用。

卷积神经网络默认输入是图像，可以让我们把特定的性质编码入网络结构，使是我们的前馈函数更加有效率，并减少了大量参数。

卷积神经网络利用输入是图片的特点，把神经元设计成三个维度 ： width, height, depth

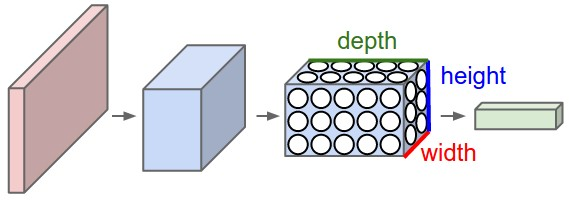


图4-1 卷积神经网络基本结构

一个卷积神经网络由很多层组成，它们的输入是三维的，输出也是三维的，有的层有参数，有的层不需要参数。

1. **卷积层**

卷积神经网路中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法优化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网络能从低级特征中迭代提取更复杂的特征。

其主要特征为：（不加以赘述）

* + - 局部感知
    - 空间排列
    - 参数共享

基于以上特点，卷积层的数据量相比于普通神经网络要小很多，并且在加深层数后能够提取到高维的图像特征。

1. **池化层**

训练时通常在卷积层之后会得到维度很大的特征，池化层能够将特征切成几个区域，取其最大值或平均值，得到新的、维度较小的特征。具体来说，池化即下采样，目的是为了减少特征图。池化操作对每个深度切片独立，规模一般为 2＊2，相对于卷积层进行卷积运算。经过池化操作后深度大小不变。

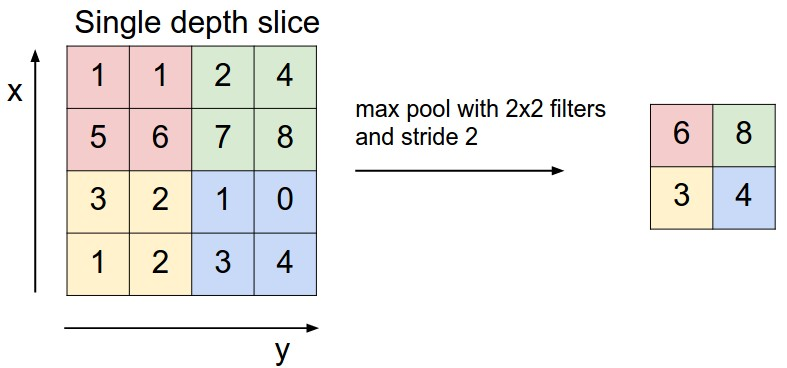


图4-2 池化层原理

1. **Dropout层**

Dropout比较简单的理解就是：我们在前向传播的时候，让某个神经元的激活值以一定的概率p停止工作，这样可以使模型泛化性更强，因为它不会太依赖某些局部的特征。当前Dropout被大量利用于全连接网络，而且一般认为设置为0.5或者0.3，而在卷积网络隐藏层中由于卷积自身的稀疏化以及稀疏化的ReLu函数的大量使用等原因，Dropout策略在卷积网络隐藏层中使用较少。总体而言，Dropout是一个超参，需要根据具体的网络、具体的应用领域进行尝试。

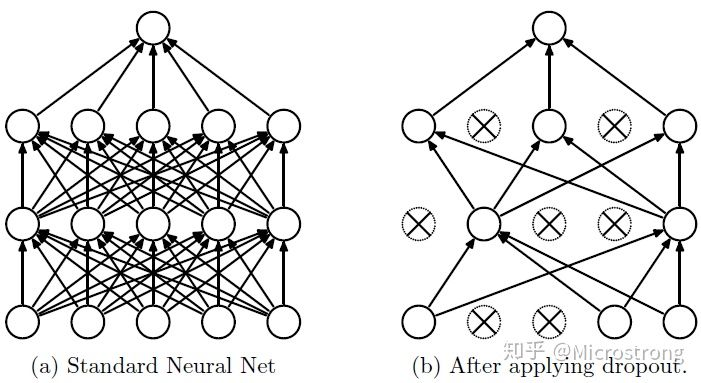
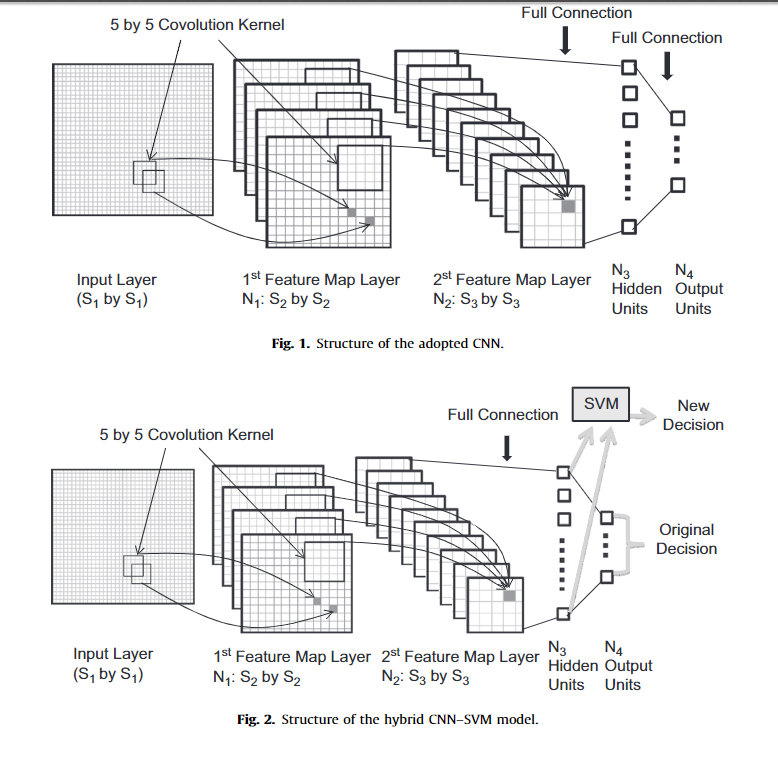


图4-3 dropout层原理

1. **CNN提取特征SVM训练相关理论支撑**
2. **CNN-SVM模型结构**

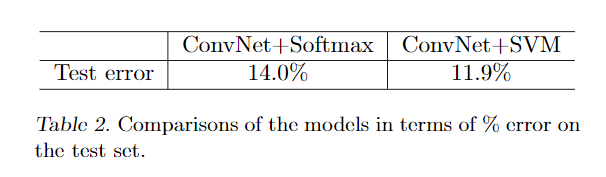


通过查阅相关资料[1]，我们查找到CNN-SVM模型的具体实例如上。CNN与混合模型CNN-SVM的区别在于最后的分类器不是采用softmax层等直接获取分类结果，而是将原有的的最后的输出分类层换为SVM进行新的决策。

1. **CNN-SVM性能**

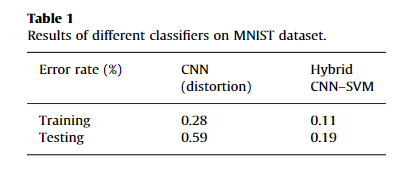
* CIFAR-10数据集分类结果

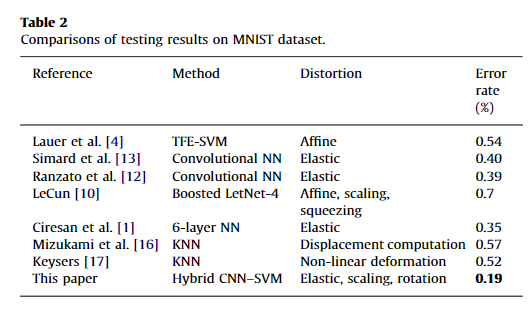
通过Yichuan Tang的实验[2]我们可以具体的实验结果数据如下：



由该表格数据来看，CNN-SVM的测试集错误率相比于CNN-SVM要低一些。

* MNIST手写数据集分类结果

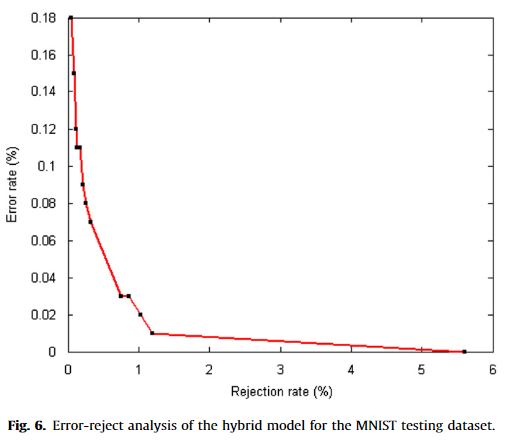




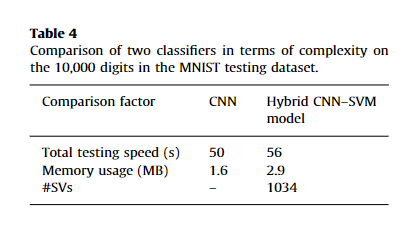
MNIST手写数据集是深度学习中十分经典的数据集，Xiao-Xiao Niu等人[1]通过比较不同实验者采用的方法发现CNN-SVM的性能要更加好。从误差率来看，和其他算法相比也有很大的提升。

1. **CNN-SVM可行性和复杂度**

为了对所提出的模型进行完整的评估，Xiao-Xiao Niu等人[1]研究了混合模型的可靠性。他们通过拒绝机制来实现可靠性分析，当排名预测中的前两个概率值之间的差小于预定义的阈值时，将拒绝测试样本。具体的实验结果如图所示：



除外，Xiao-Xiao Niu等人[1]还利用测试速度，内存占用和SVs的数量来对CNN-SVM的复杂度进行分析，从下表可以看出CNN-SVM相比于CNN牺牲了一定的内存和测试速度。



1. **参数设定与调整**
2. **参数调试过程**

* **模型具体说明**

本次实验的模型是CNN-SVM混合模型。

整个训练过程:

1.利用本征脸还原原始图像数据

2.利用还原的图像数据集训练CNN分类器，并加以保存。

3.观察分类器性能大体上是否符合要求，是否存在过拟合、准确率不高等问题，若存在则返回步骤1.

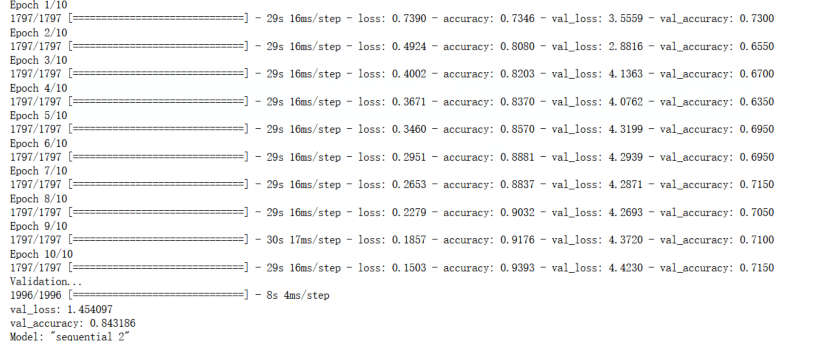
4.将训练集和测试集通过CNN模型计算，返回全连接层得出的参数，此时得到的就是用于SVM分类的特征数据集

5.通过调节SVM参数，得到SVM分类器，观察准确率、召回率、查准率等指标。若存在过拟合、欠拟合等问题，则重新分析模型问题再做修改。

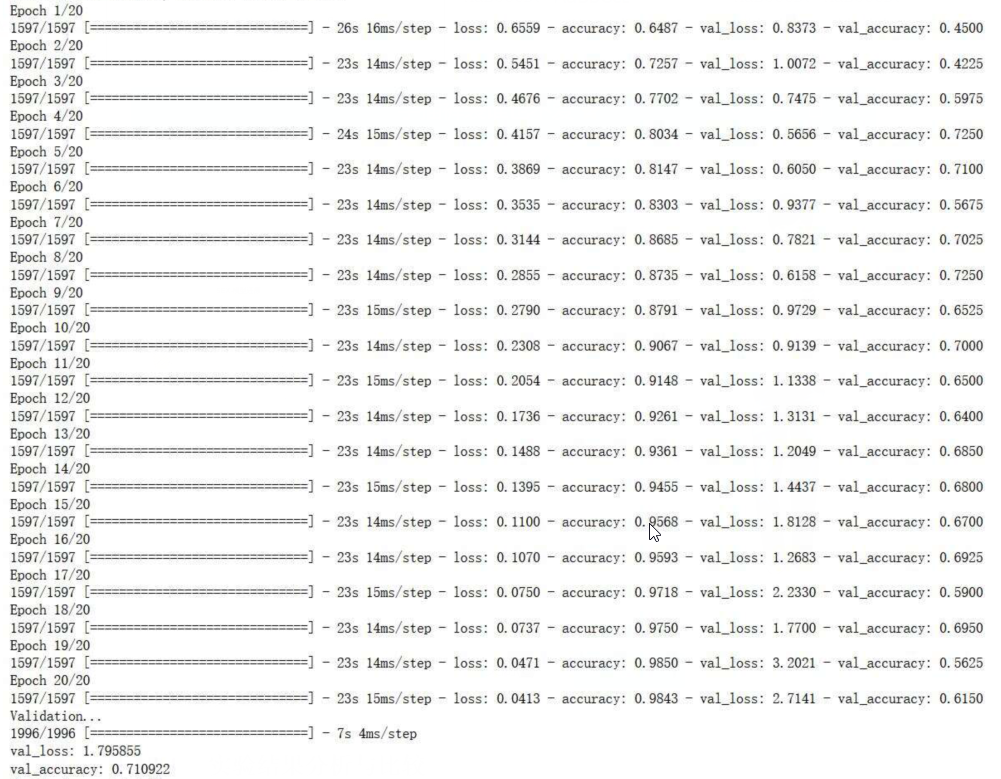
* **CNN层搭建与选择**

1.**训练epoch选择**

Epoch=10



Epoch=20



从以上两种结果可以很明显的看出无论是10还是20都是偏大的,都出现了比较严重的过拟合情况,尤其是epoch=20时.因此,epoch选择要小于10,取epoch=5.

**2.防止过拟合Dropout和BatchNormalization层选择**

从上面的结果可以看出存在一定的过拟合情况,因此可能需要增加相应的dropout层或者batchnormalization层以防止过拟合情况的出现。

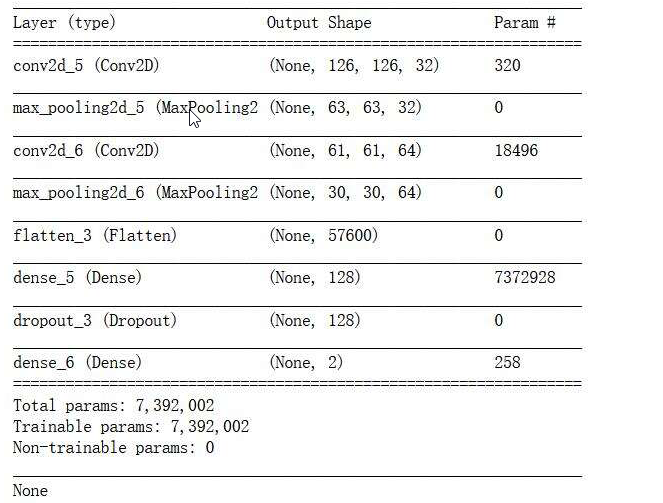


图 带dropout层卷积神经网络结构

由于数据本身量并不够多,加上此次设计的网络较为简单和基础,因此dropout层的防止过拟合的效果并不明显,经过训练之后准确率有下降的趋势,因而不加以选择.

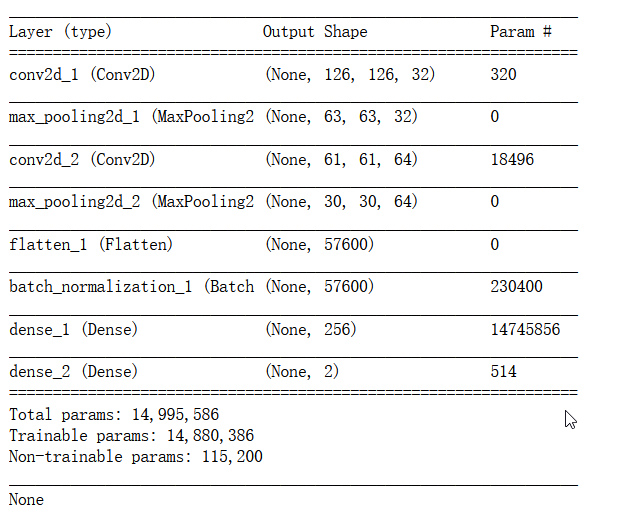


图 带层BatchNormalization卷积神经网络结构

BatchNormalization层,在深度学习种也会使用到一种防止过拟合的方法。但是在此次实验中效果和dropout相同也没有很好的一个效果。基于此，从大致一个方向来看，是所使用的数据没有处理好，导致CNN提取特征并加以预测的效果也没有达到预期的效果。

* **SVM调试参数选择**

根据传统的算法进行分析，该数据基本上可以确定在{'C': 10, 'kernel': 'rbf'}这一个参数的附近，可以依照经验调节惩罚因子得到训练结果。从以下训练结果，我们可以很清楚的看出过拟合情况十分的严重，模型的泛化能力十分差。训练准确率在达到93.9%和98.7%的情况下，对于测试集的预测结果也不过72%，78%。

结合CNN的调试过程，我们可以很大程度上将效果较差的情况“归咎于”数据本身的问题上。



* **数据再处理**

通过比较训练集和测试集，可以清楚的看到，两者数量差不多接近均在2000左右。但是就整体而言，训练集数量相对较少了，并不能够很好的去训练出一个比较好的分类器；相对的，测试集较整体而言占的比例相对较大。因此，对数据需要重新的整合和划分。





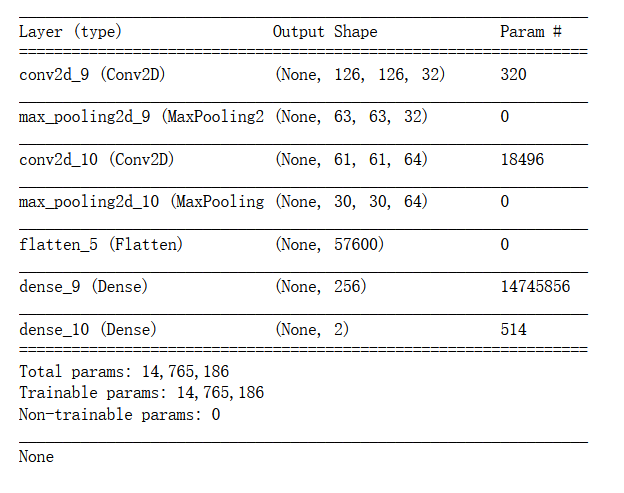
1. **模型参数确定**

* **CNN参数**

整体模型：



性别预测模型：



年龄预测模型：



* **SVM参数**

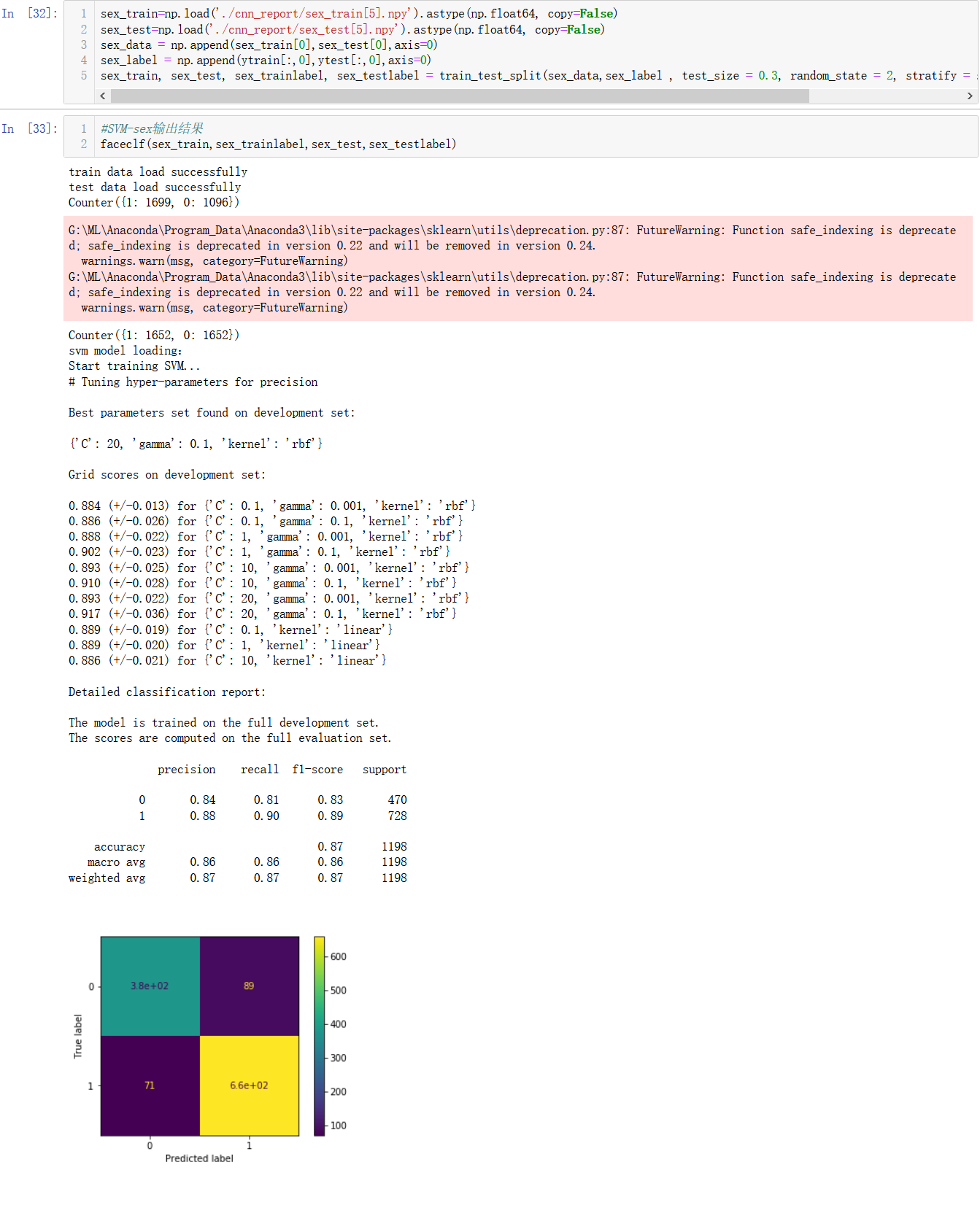


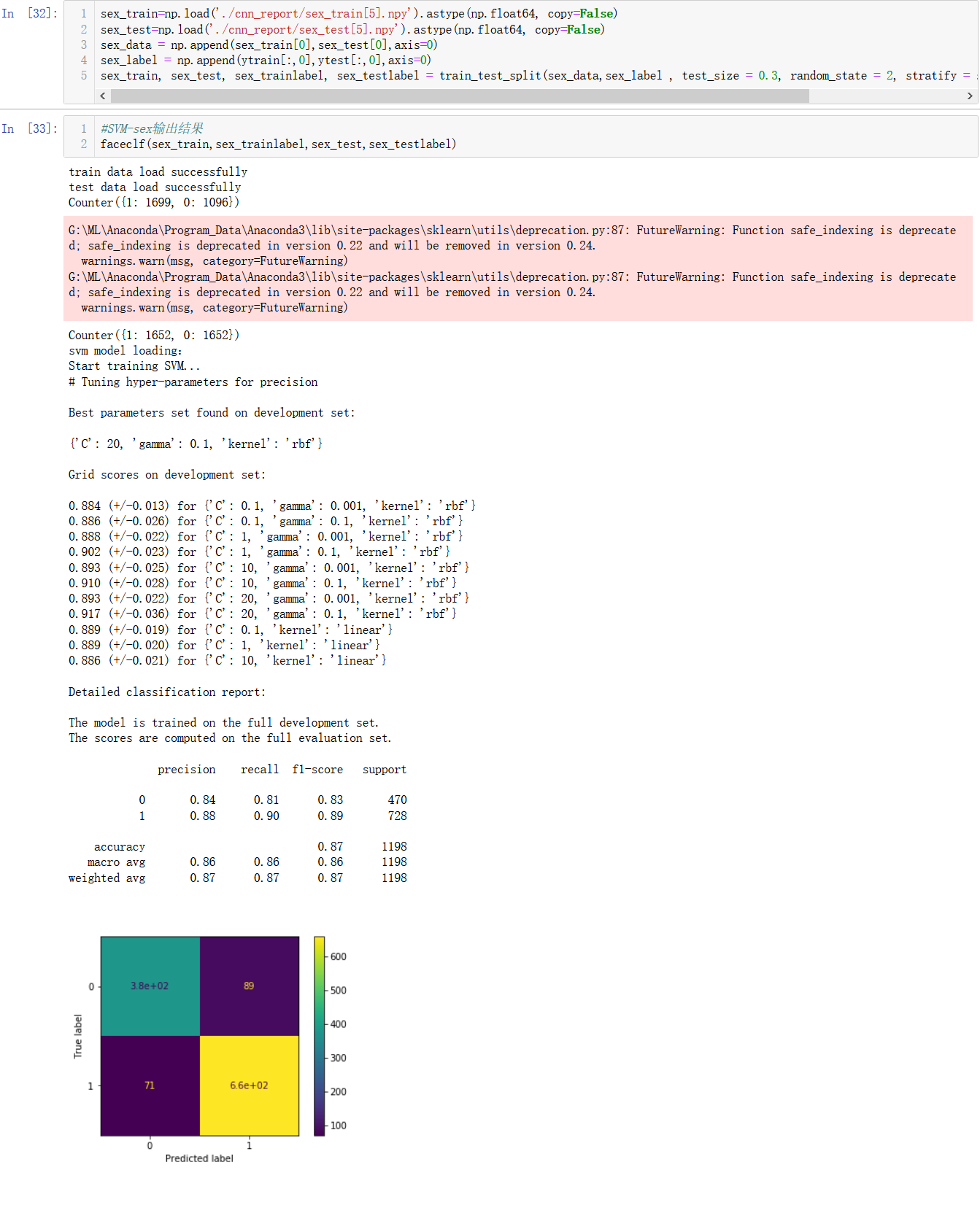
1. **实验结果分析与比较**
2. **性别结果**

* CNN训练结果



* SVM分类结果



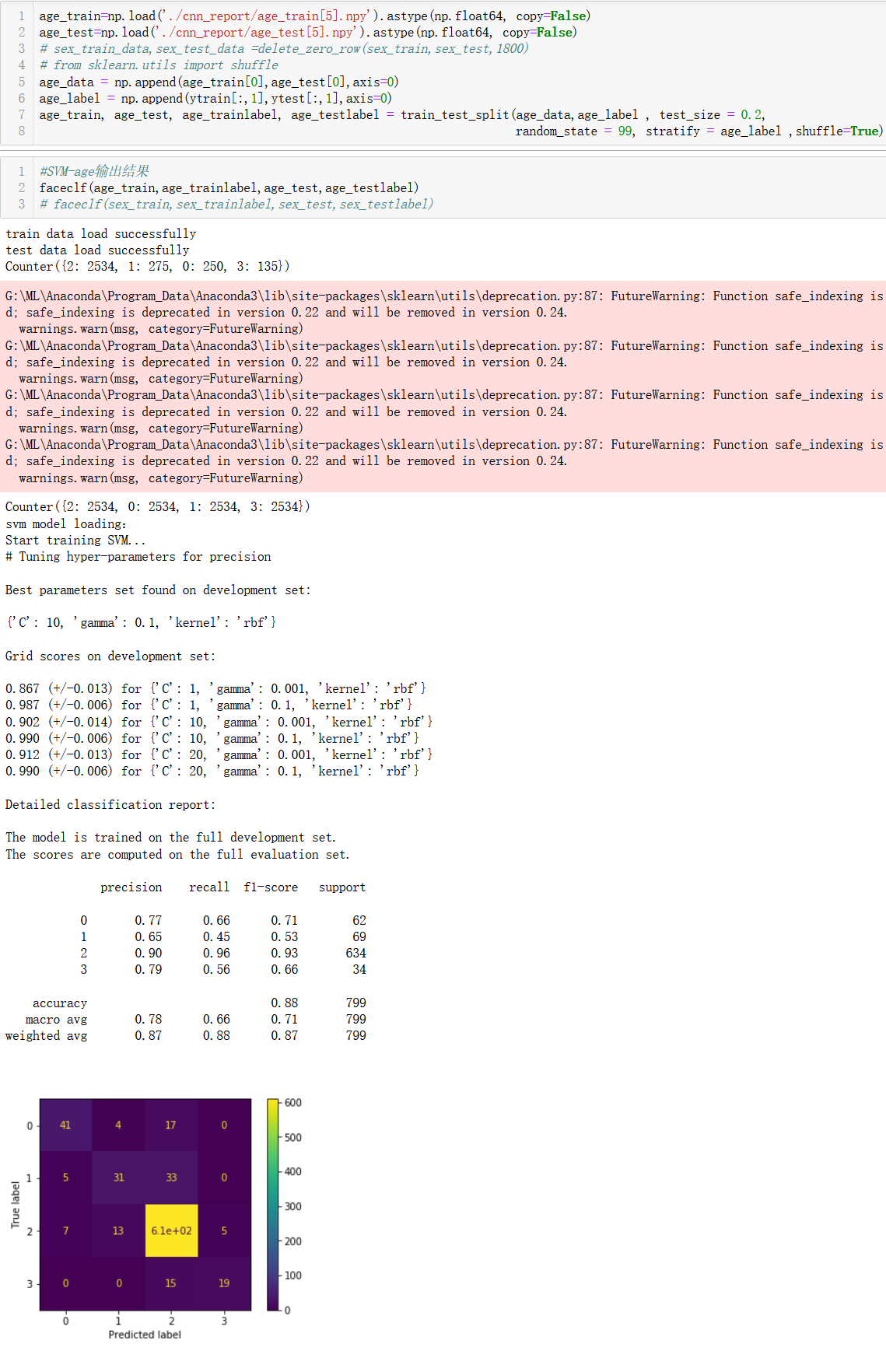
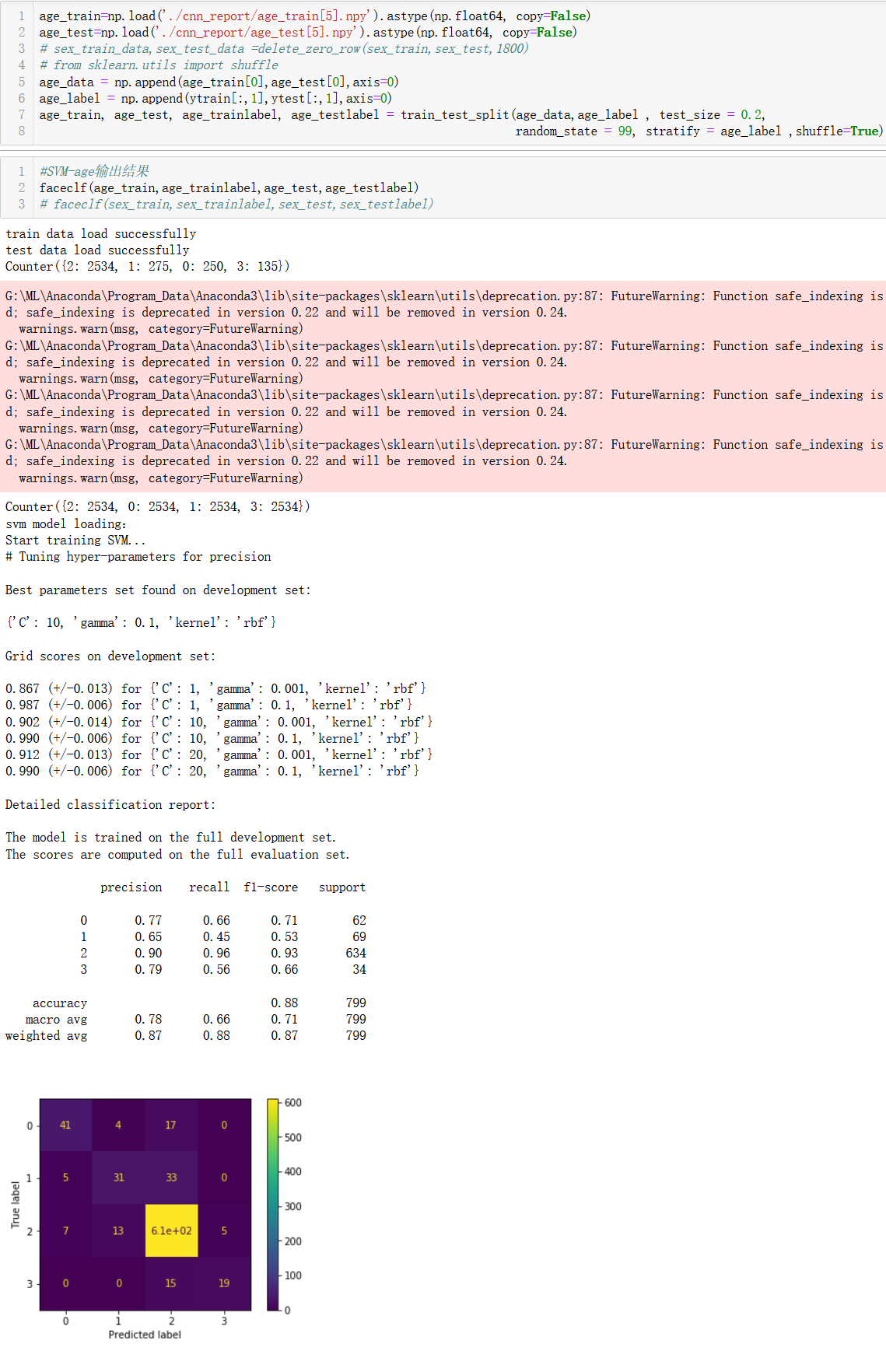


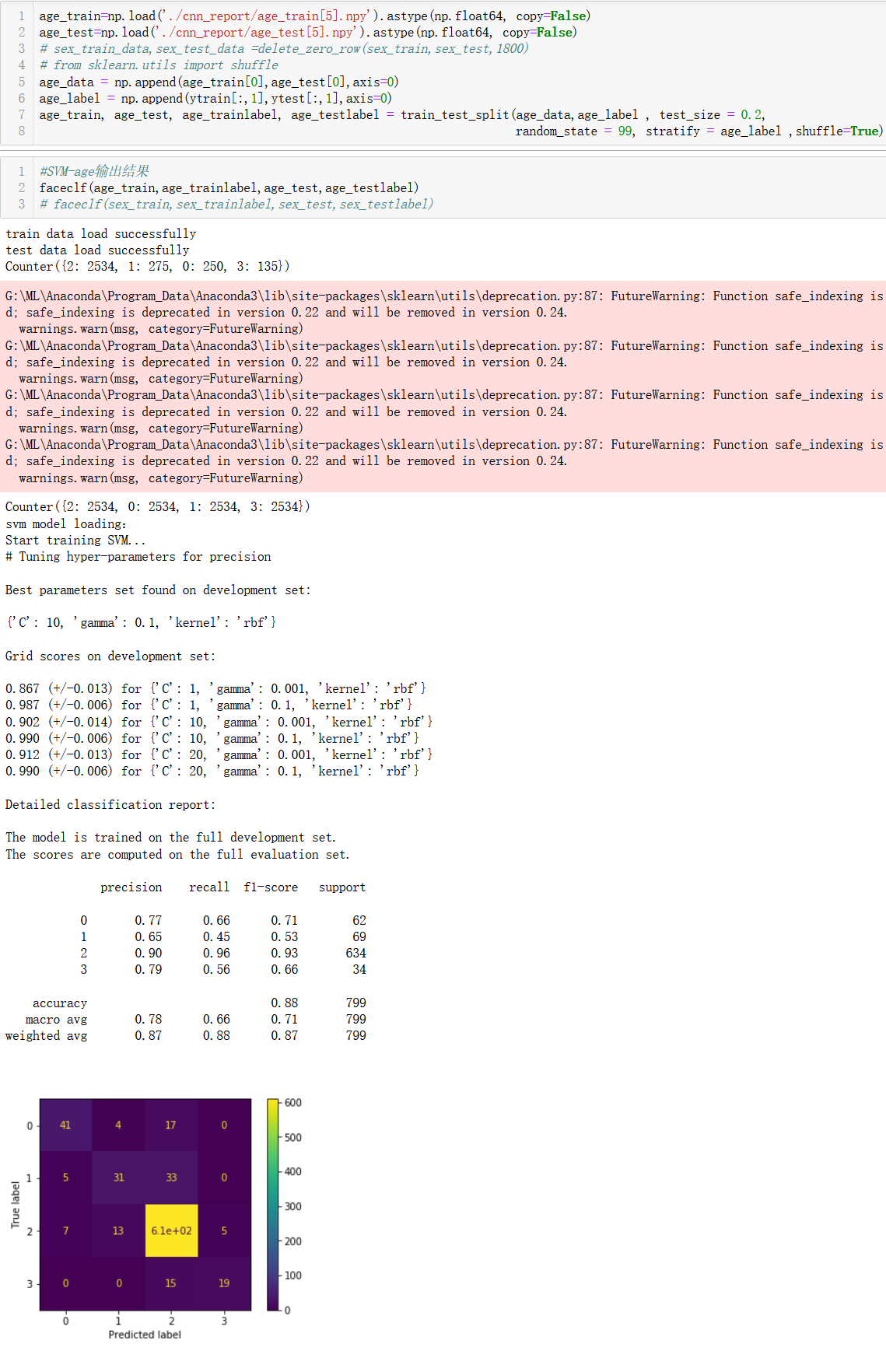
1. **年龄结果**

* CNN训练结果



* SVM分类结果：

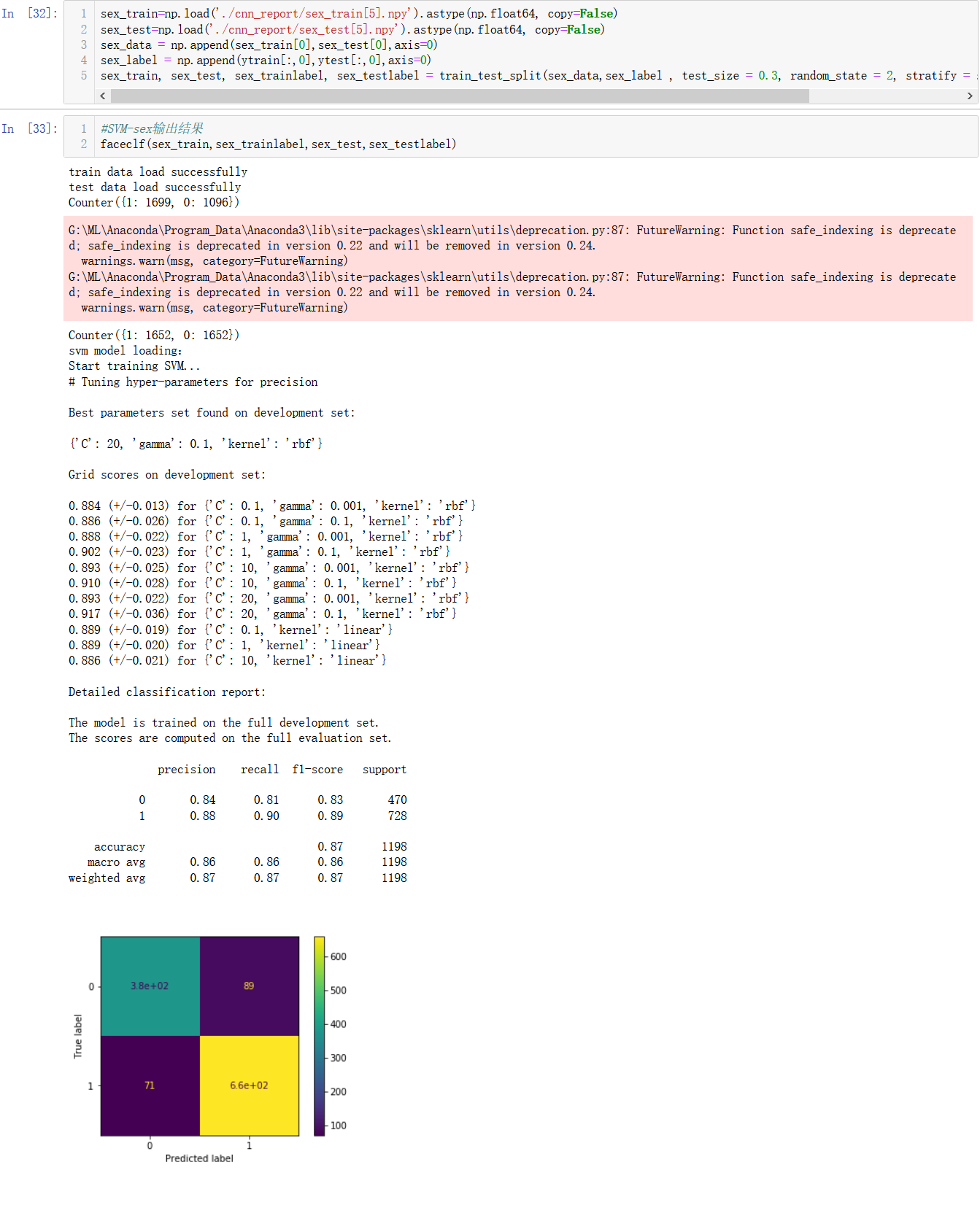




1. **结果分析与比较**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prediction | | | | |
|  | Precision | Recall | F1-score | support |
| 0-female | 0.85 | 0.80 | 0.82 | 313 |
| 1-male | 0.88 | 0.91 | 0.89 | 486 |
| accuracy |  |  | 0.87 | 799 |
| Marco avg | 0.86 | 0.85 | 0.86 | 799 |
| weighted avg | 0.87 | 0.87 | 0.87 | 799 |
| Best parameters | | | | |
| {'C': 20, 'gamma': 0.1, 'kernel': 'rbf'} | | | | 0.917 (+/-0.030) |

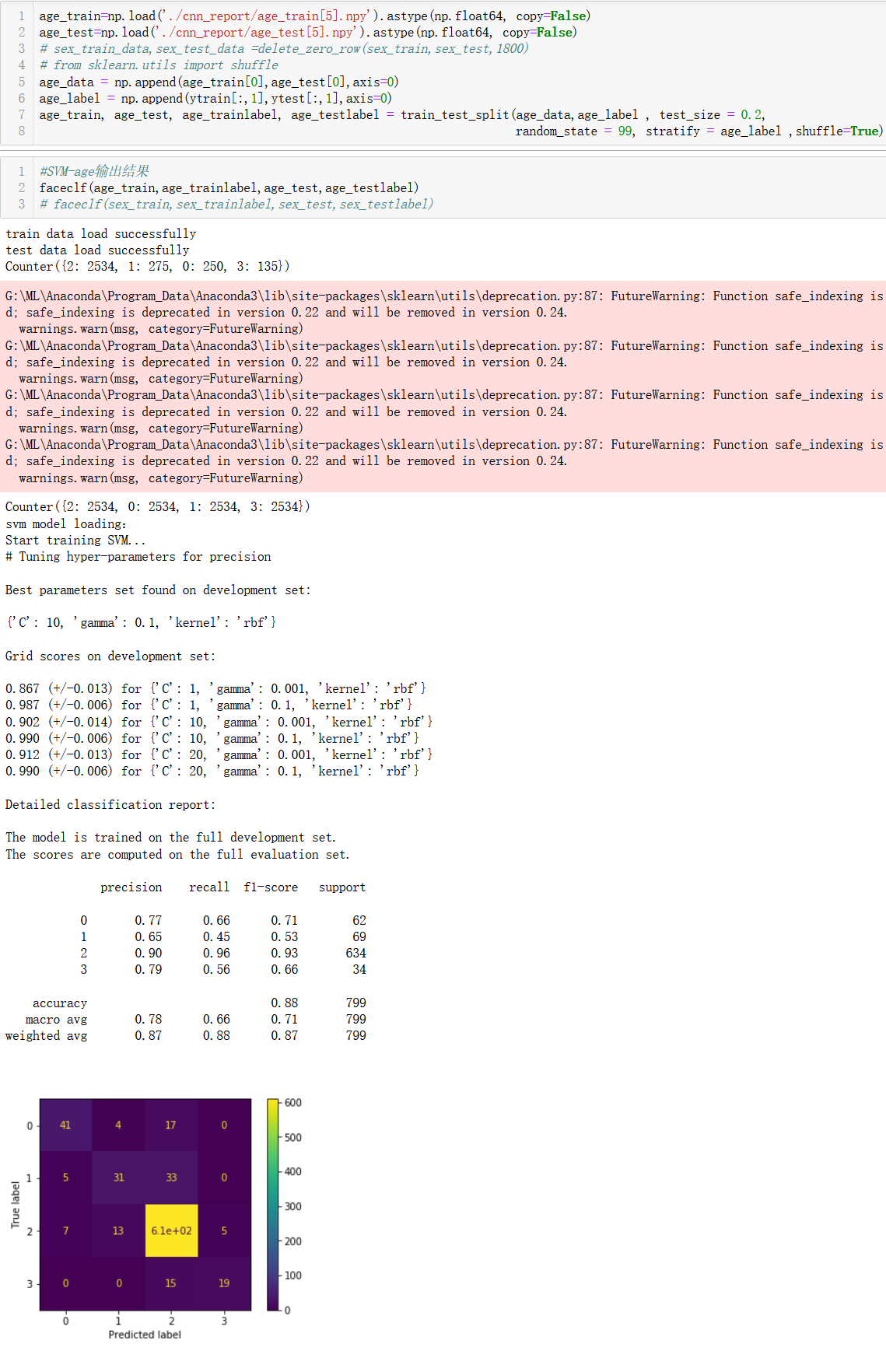
表 性别分类结果



从上表数据我们可以看到分类器对于测试集样本的准确率和训练准确率大体上一致（训练集：91.7% 测试集：87%）。并且从混淆矩阵和其对应的查准率、召回率都挺高，表示该分类器有很好的性能。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prediction | | | | |
|  | Precision | Recall | F1-score | support |
| 0-child | 0.77 | 0.66 | 0.71 | 62 |
| 1-teen | 0.65 | 0.45 | 0.53 | 69 |
| 2-adult | 0.90 | 0.96 | 0.93 | 634 |
| 3-senior | 0.79 | 0.56 | 0.66 | 34 |
| accuracy |  |  | 0.88 | 799 |
| Marco avg | 0.78 | 0.66 | 0.71 | 799 |
| weighted avg | 0.87 | 0.88 | 0.87 | 799 |
| Best parameters | | | | |
| {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'} | | | | 0.990 (+/-0.006) |

表 年龄分类结果



结合训练结果来看，我们的训练在一定程度上存在过拟合的情况，因为训练集训练准确率可以达到99%，但是测试数据才88%，相差了11%的精度。通过表 我们可以看到其实整体的出现过拟合的原因还是由于adult类的占比相对大得多，其他占比相对较低导致的。但就整体而言，分类器的效果在一定程度上也还算过得去，查准率、召回率和F1值还是再一定程度上反映了实际的情况。

但从混淆矩阵来看，分错的大概率都是adult类，那么可以得到的一个结论就是模型的过拟合大部分是由于将其他类别错分到adult中。结合我们的数据来看，每个类别比例相差比较大，SMOTE算法平衡后的数据并不能完全替代真实数据，所以出现这种情况是在预料之中的。

1. **CNN-SVM算法总结**

结合以上的数据，从识别精度和可靠性两个方面来看，混合CNN-SVM模型有很好的一个效果。就整体而言，该模型受到数据影响十分严重。从性别分类来看，训练准确率和测试准确率基本没差别，并且从召回率、查准率和F1值来看也有相对较好的性能。但从年龄分类来看，很明显的可以看出受数据类别本身不均衡的影响，类别间数据量相差十分之大，利用SMOTE算法只能够在一定程度上减少这些影响并不能够完全消除这些影响。

通过比较可以很明显的看出，两个分类任务都存在一定程度的过拟合，性别分类的过拟合基本是可以容忍和接受的。但是年龄分类任务的过拟合程度是很明显的，结合召回率、查准率、F1值和混淆矩阵，不难看出模型依旧陷入了“多类优先”的困境。

如果需要解决以上问题首要的需要对数据进行增强，不是利用SMOTE算法去实现拟合，而是在原有的图像基础上进行旋转，放缩等手段增加少类别样本的数目。除此之外，可以对图像数据进行一定的操作，如增强纹理的区别利用传统的图像识别算法，使得图像特征更加明显。

在解决了数据问题之后，分类器还能够有所改进，通过增加卷积神经网络的层数可以有效地提高图像高维特征的提取，但这种方法必然会带来更多内存的占用和一定的过拟合。

1. **SVM算法和CNN-SVM算法比较**
2. **性能分析**

在4.2.2中我们就已经引用Xiao-Xiao Niun等人[1]的实验数据，可以清楚的看到CNN-SVM较CNN本身有一定提高，跟其他类型算法相比也不逊色。

回到本次实验中，结合章节3和4，我们可以清楚的看到后者的性能在一定程度上更加突出。这不仅仅是训练集的训练准确率，而更多的体现在召回率、查准率、F1值这些更加综合的指标。通过混淆矩阵可以看出，混合模型CNN-SVM有着更佳的性能，虽然本身也存在过拟合的情况，但和SVM算法相比过拟合程度相对低一些。

本次实验本身是有了特征数据的提供，显得前者使用起来其实并不算复杂。当在实际情况下，我们需要自己重新再去提取特征，进行训练，这样下来工作量也不小。那么后者CNN-SVM模型则能够利用CNN自身的特性简化这一过程，并且效果与传统的特征提取算法相比也不差。

1. **复杂程度分析**

在本次实验中，我们SVM算法用的是已经经过PCA降维的特征数据，相比于混合模型CNN-SVM在数据上肯定是简易很多的。因为后者，我们需要先经过还原数据，本身的特征数据占用并不大，但当还原成原始图像数据内存占用比之前要远大的多。

其次，由于CNN处理的是原始图像数据，那么其包含的数据量相比于之前必然要大得多，因此不难得出和Xiao-Xiao Niun等人[1]一样的结论，CNN-SVM不仅增大了使用内存而且会使得检测速度会相应下降。

1. **应用的范围**

SVM算法，本身其实就在各类任务中有很好的表现，并且由于本身的简洁易操作，应用范围本来九十分广泛。混合模型CNN-SVM算法，结合了CNN和SVM两者的长处，虽然性能有一定的提高，但是应用范围则变为CNN常用的图像领域。

因此，在具体情况下，我们一方面需要考虑分类器性能，另一方面也要顾虑到整体系统的复杂程度，综合这两种情况做出选择。

经典算法的原理、参数设定与调整、实验结果的分析比较、结论

智能算法/改进算法的原理、参数设定与调整、实验结果的分析比较（必须要有与经典算法的结果对比）、结论

eijkel2

图 2-1 曲线图.

3 B算法

另一类算法[[1](#_ENREF_1)].

3.1 基本原理

原理

表 3.1-1 衡量指标

| 名称 | 含义 |
| --- | --- |
| F1 |  |
| F2 |  |

继续。

*3.1.1 子标题*

公式

(3.1.1-1)

继续。

4 C算法

5 结论

内容要求：

做了什么，取得了什么结论（同种算法之间？不同种算法之间？）

需要改进的方面

# References:

[1]. Xiao-Xiao Niun, C.Y.S., A novel hybrid CNN – SVM classifier for recognizing handwritten digits. 2011.

[2]. Tang, Y., Deep Learning using Linear Support Vector Machines. 2013.

附录

表 A1 成员分工及贡献度自评

| 姓名 | 个人分工 | 贡献度 |
| --- | --- | --- |
| 江熠铧 | 具体内容，课程报告第1、2节 |  |
| 蔡瑞涵 | 具体内容，课程报告第3节 |  |
| 谭美健 |  |  |
|  |  |  |

课程体会与建议

具体内容