北京石油化工学院
BEIJING INSTITUTE OF PETROCHEMICAL TECHNOLOGY

实验报告

实验名称：Sklearn 与 MLPP 的 SVM 实例研究

班级：智研 211

姓名：王利猛

学号：2021540044

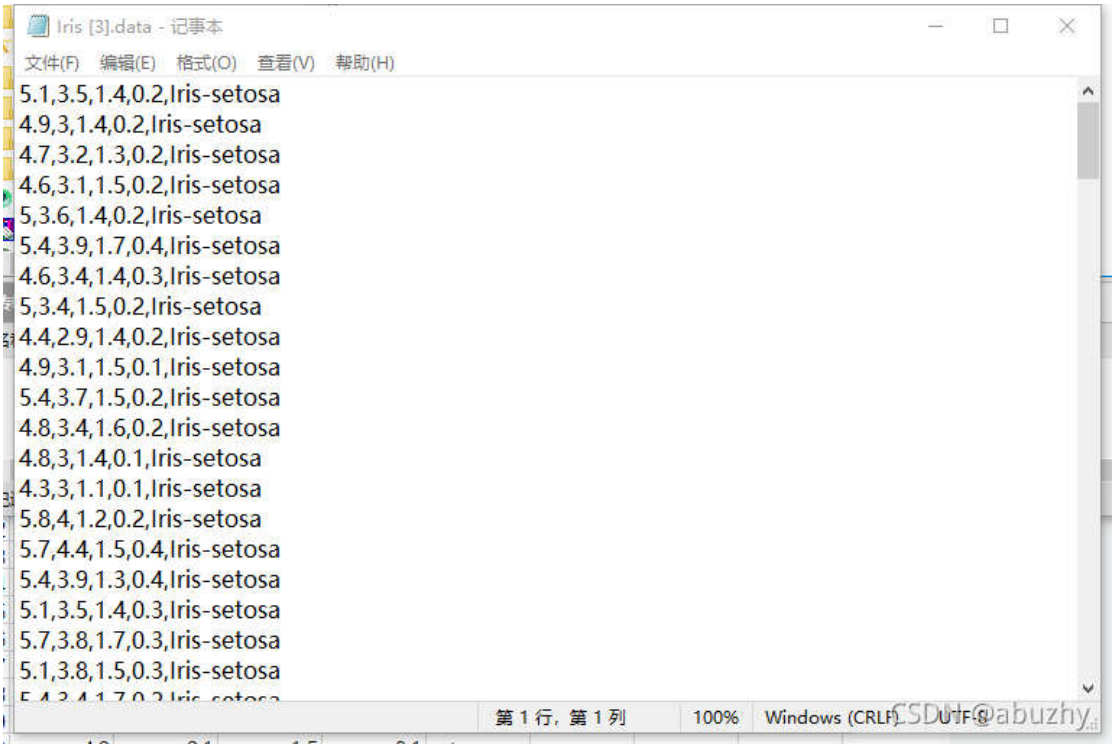
教师：徐文星

人工智能研究院

目录

一、数据集介绍

Iris 鸢尾花数据集是一个经典数据集，在统计学习和机器学习领域都经常被用作示例。数据集内包含 3 类共 150 条记录，每类各 50 个数据，每条记录都有 4 项特征：花萼长度、花萼宽度、花瓣长度、花瓣宽度，可以通过这 4 个特征预测鸢尾花卉属于（iris-setosa, iris-versicolour, iris-virginica）中的哪一品种。下图给出数据集前 20 条数据。

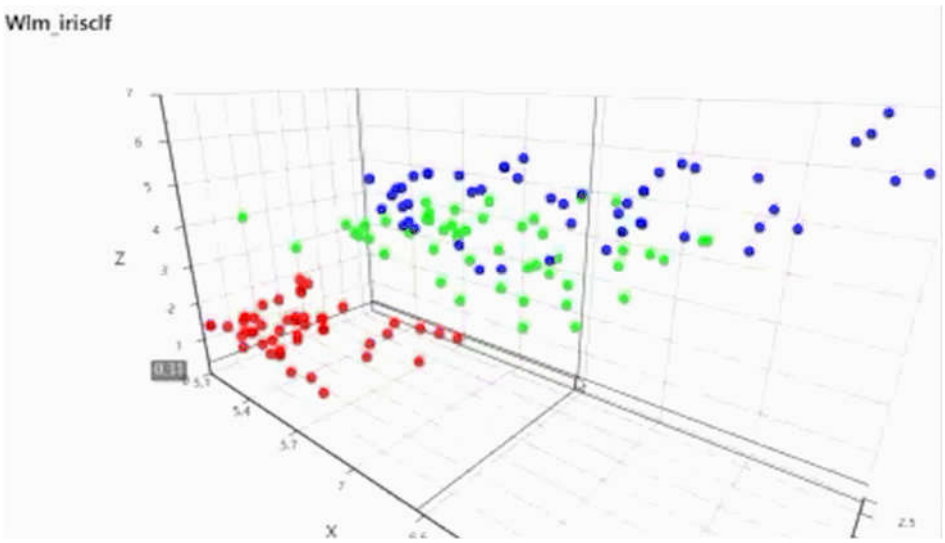


数据集获取地址：

https://archive.ics.uci.edu/ml/machine-learning-databases/iris/

1.1 数据集可视化：

我使用的是 pyecharts 对 sklearn 中的鸢尾花数据进行可视化展示。用数据集前三个参数作为 xyz 坐标，可以大致看出这些数据点在空间中的分布。在下图的 gif 中，使用鼠标点击每个特征向量，即可显示出所属类别（红绿蓝代表三种鸢尾花）和前三个特征值（xyz 坐标）。pyecharts 需要在使用前用匹配 pip 手动安装，安装完之后可以直接使用。



二、实验目标

2.1 Sklearn 的 svm.SVC 模型调用

先将 label 与数字进行对应，简化数据集。建立如下对应关系：

Iris-setosa: 0, Iris-versicolor: 1, Iris-virginica: 2

切分输入数据和输出数据后（数据集前四列为输入数据，最后一列为输出）将数据集打乱。再用 train_test_split 函数以 7:3 的比例随机划分到训练集与测试集。利用 Sklearn 的 svm.SVC 模型进行拟合，用 score 函数计算训练后的模型在训练集和测试集上的准确率。输出 svm 对测试集的预测值与真实值，混淆矩阵和 svm 的决策函数。为了更形象地描述 svm 的分类边界，我将 150 条数据集的前两个特征（花萼长度、花萼宽度）散点图和分类边界在该平面上的投影也输出到了同一张图里。

2.2 修改超参 C 与 gamma 值

对于 SVM 模型，最主要的参数就是 C 和 gamma，C 被称为惩罚因子，越大说明对错误的容忍度越小。太大可能发生过拟合，太小可能会导致容错率过高，这样的模型就没有意义了。

对于 C 和 gamma 参数的确定，一般是通过交叉验证的方法来进行确定的。这样的话就能够得到不同（C，gamma）组合下模型的得分，取得分最高的那组（C,gamma）即可。如果碰到有相同分的，一般考虑参数 C,取 C 小的。若 C 也相同，则取先出现的那一组 gamma。

2.2.1 小节给出了双循环的交叉验证算法自编程实现，只能输出每个 k-折验证的最大准确率及超参 C 和 gamm 最优选择范围。2.2.2 小节结合 GridSearchCV 方法准确输出 C 和 gamma 的最优值及相应准确率。

2.2.1 交叉验证的实现

交叉验证最基本的方法被称之为 k-折交叉验证 。k-折交叉验证得出的性能指标是循环计算中每个性能指标（如在测试集上的准确率）的平均值。代码里用 RepeatedKFold（进行 k-折交叉验证）和 cross_val_score（计算交叉验证分数）函数进行实现。该方法虽然计算代价很高（与 K 和 fold 有关），但是它不会浪费太多的数据（没有验证集）。

本小节给定 C 与 gamma 的取值范围，输出在此二维网格范围中的不同的 K 与 fold 的交叉验证性能指标最大值。并绘制出 C，gamma 与该最大值的三维模型。

2.2.2 网格搜索算法

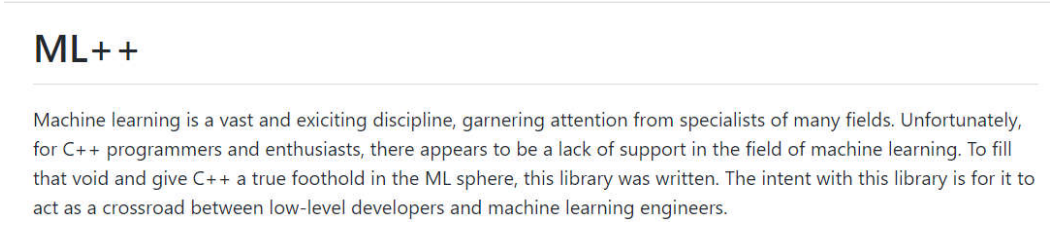
与 2.2.1 小节算法类似。本小节也是将(C,gamma)组成了一个二维网格，再与性能指标最大值组成三维模型。不同点在于直接调用 sklearn 的 GridSearchCV 函数，输出最优的 C 与 gamma 以及对应的最大值。从而实现参数的最优选择。

GridSearchCV 缺点在于遍历所有可能参数的组合，在面对大数据集和多参数的情况下，非常耗时。所以这个方法适合于小数据集，一旦数据的量级上去了，很难得到结果。

2.3 C++机器学习库 MLPP 的 SVC 实现

C++机器学习库 MLPP 是针对 C++设计的机器学习库。已于 github 上开源。地址为：https://github.com/novak-99/MLPP。由于这个库至今仍在不断更新，所以与 sklearn 相比功能还不是很完善。截止到本文提交，支持向量机部分仅支持支持向量二分类问题（SVC）。

本小节将以该库提供的数据集 BreastCancer(数据一共有 569 组 30 维。其中两个分类分别为良性（benign）357；恶性（malignant）212。代码中以 1 代表良性，-1 代表阴性)为例，测试它的分类效果。



三、参数设置

3.1 Sklearn 的 svm.SVC 模型调用

	C	kernel	gamma
第一次	1	rbf	10
第二次	5	linear	8

注：

kernel='linear'时，为线性核，C 越大分类效果越好，但有可能会过拟合（defaul C=1）。

kernel='rbf'时（default），为高斯核，gamma 值越小，分类界面越连续；gamma 值越大，分类界面越“散”，分类效果越好，但有可能会过拟合。

3.2 修改超参 C 与 gamma 值

3.2.1 交叉验证的实现

	C	kernel	gamma	K	Fold
五次五折	1-15	rbf	1-11	5	5
八次十折	1-15	rbf	1-11	8	10
五次五折	5-15	rbf	1-11	5	5

3.2.2 网格搜索算法

	C	kernel	gamma	K	Fold
五次五折	1-15	rbf	1-11	5	5
八次十折	1-15	rbf	1-11	8	10
五次五折	5-15	rbf	1-11	5	5

3.3 C++机器学习库 MLPP 的 SVC 实现

C++程序主函数如图，数据集还是以 7:3 进行划分。运行环境 Linux 20.04LTS：

```
int main() {

    // // OBJECTS
    Stat stat;
    LinAlg alg;
    Activation avn;
    Cost cost;
    Data data;
    Convolutions conv;

    // SUPPORT VECTOR CLASSIFICATION
    auto [inputSet, outputSet] = data.loadBreastCancerSVC();
    SVC model(inputSet, outputSet, 1);
    model.SGD(0.00001, 100000, 1);
    alg.printVector(model.modelSetTest(inputSet));
    std::cout << "ACCURACY: " << 100 * model.score() << "%" << std::endl;

    return 0;
}
```

四、结果分析

4.1 Sklearn 的 svm.SVC 模型调用

第一次实验：C=1，kernel=rbf，gamma=10

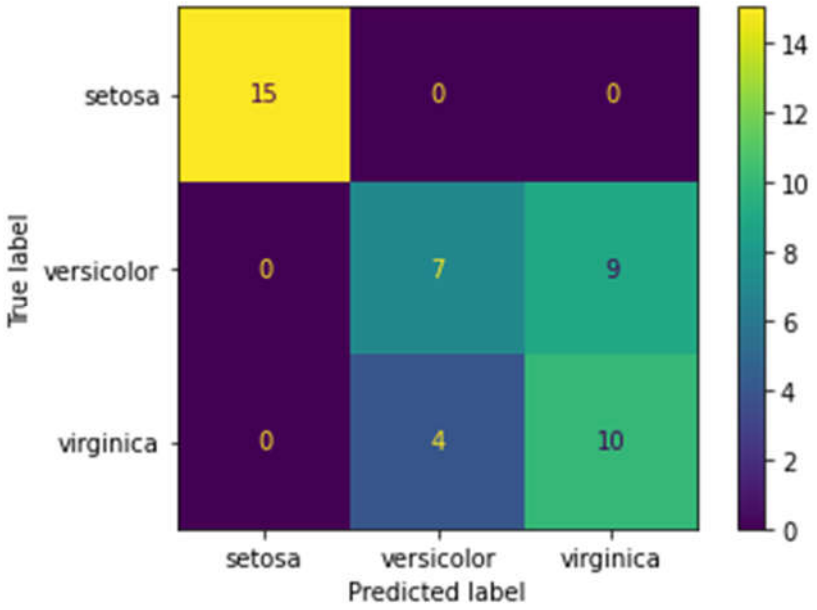
直接使用训练集对 svm.SVC 模型进行拟合训练，得到训练后的模型对训练集和测试集 data 预测正确率如下图：

```
训练集正确率：  0.8571428571428571
测试集正确率：  0.7111111111111111
测试集真实值:
 [2. 1. 1. 0. 2. 1. 1. 2. 0. 2. 0. 2. 1. 0. 2. 2. 2. 1. 2. 1. 2. 2. 0. 1.
 1. 2. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 2. 1. 0. 2. 0. 1. 0. 1.]
测试集预测值:
 [2. 2. 2. 0. 2. 1. 2. 1. 0. 2. 0. 1. 1. 0. 2. 2. 1. 2. 1. 2. 2. 2. 0. 2.
 1. 2. 0. 0. 0. 2. 0. 0. 0. 1. 1. 0. 2. 2. 1. 0. 2. 0. 1. 0. 2.]
```

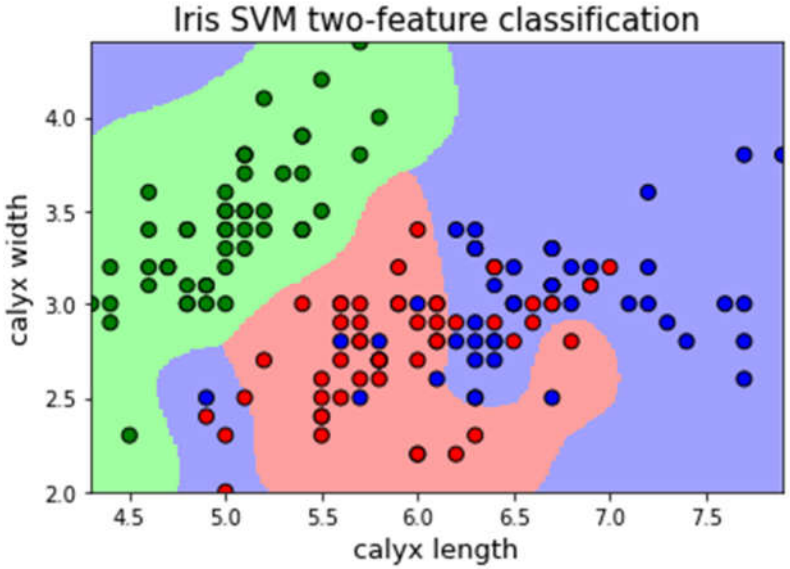
以下是训练后模型的决策函数（部分）。每个数字代表模型的权重值。

```
decision_function:
 [[ 2.22910515 -0.18590652  0.838745  ]
 [-0.22242179  2.19983866  1.1123515 ]
 [ 2.22881417 -0.18269263  0.83532031]
 [-0.22858691  2.22857628  1.00010761]
 [-0.22294887  1.0000952   2.22293842]
 [-0.22626201  2.22223833  1.03377645]
 [-0.16930523  0.83605898  2.22222188]
 [ 2.22221057 -0.17899278  0.8478301 ]
 [-0.21882244  2.20087806  1.09427475]
 [ 2.22881417 -0.18269263  0.83532031]
 [-0.22390613  2.22283149  1.00959059]
 [ 2.22541946 -0.18384234  0.84596544]
 [ 2.22220798 -0.17886194  0.84765703]
 [-0.20675622  0.91040933  2.22225937]
 [-0.22395186  1.11906442  2.19955915]
```

相应地可以画出测试集数据的混淆矩阵如下：



以下是数据集 150 条数据的前两项特征散点图与超平面在该平面投影图结合。可以很清晰地看见尽管决策空间仍有一些不属于此类的特征向量。但是大部分特征向量都得到了很好的划分。



第二次实验：C=5，kernel= linear，gamma=8

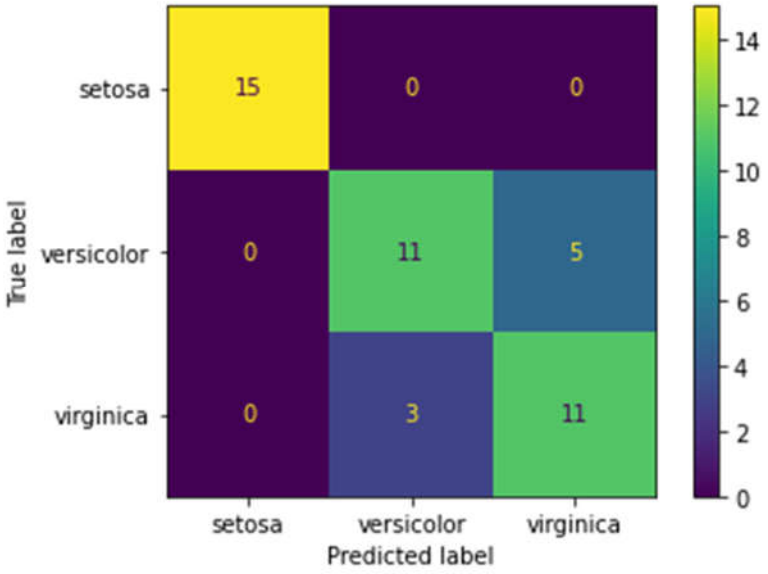
模型在训练集和测试集上的准确率如下：

```
训练集正确率：  0.8
测试集正确率：  0.8222222222222222
测试集真实值:
 [2. 1. 1. 0. 2. 1. 1. 2. 0. 2. 0. 2. 1. 0. 2. 2. 2. 1. 2. 1. 2. 2. 0. 1.
 1. 2. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 2. 1. 0. 2. 0. 1. 0. 1.]
测试集预测值:
 [2. 2. 2. 0. 2. 1. 1. 1. 0. 2. 0. 1. 1. 0. 2. 2. 2. 2. 1. 1. 2. 2. 0. 1.
 1. 2. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 2. 2. 1. 0. 2. 0. 1. 0. 2.]
```

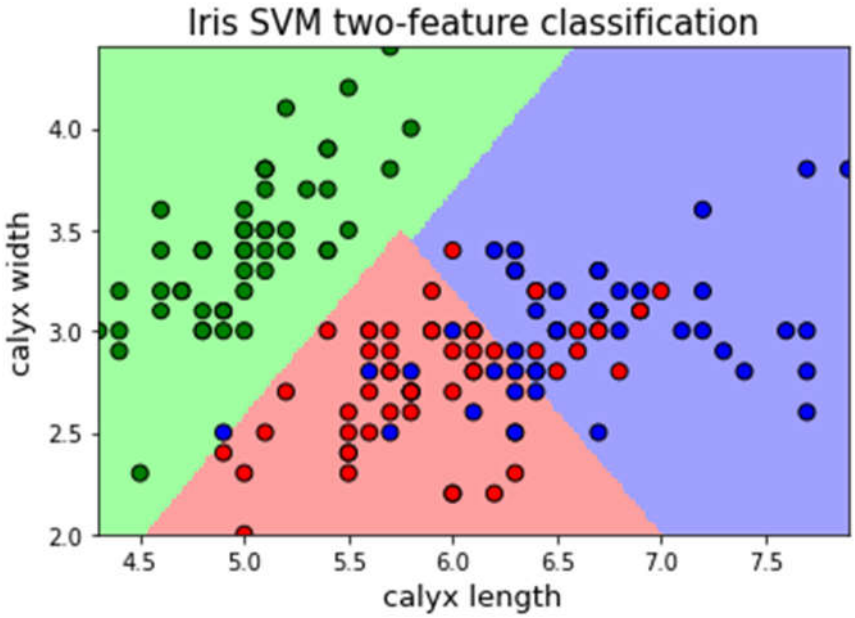
决策函数系数（部分）

```
 1. 2. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 2. 2. 1. 0. 2. 0. 1. 0. 2.]
decision_function:
 [[ 2.26994795  0.78354006 -0.23548828]
 [-0.28027115  2.25823422  1.21610109]
 [ 2.26189088  0.97231286 -0.26047637]
 [-0.26459819  2.25429407  1.12911241]
 [-0.283145    1.24511437  2.24704849]
 [-0.27144309  2.26177182  1.1404189 ]
 [-0.30820896  1.25832787  2.29940009]
 [ 2.27400603  0.81873569 -0.25803034]
 [-0.27588506  2.27066155  1.10865555]
 [ 2.26189088  0.97231286 -0.26047637]
 [-0.27408131  2.25430561  1.19489197]
```

测试集混淆矩阵：



150 条数据前两个特征的与超平面投影图：



以上数据表明，核函数的选择会影响决策边界形状。实验一的训练集准确率高于实验二，但是实验二测试集的准确度却高于实验一。这一点产生的原因很有可能是因为实验一 C 取值太小（C=1）导致模型的容错率过高，泛化性能不好。

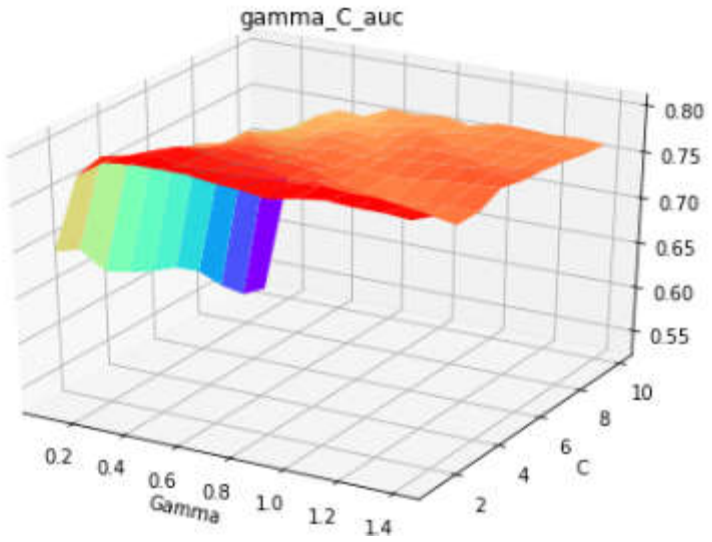
4.2 修改超参 C 与 gamma 值

4.2.1 交叉验证的实现

绘制图像如下，图中红色部分则为性能指标最大的地方，取该处的（C，gamma）作为参数进行模型训练。

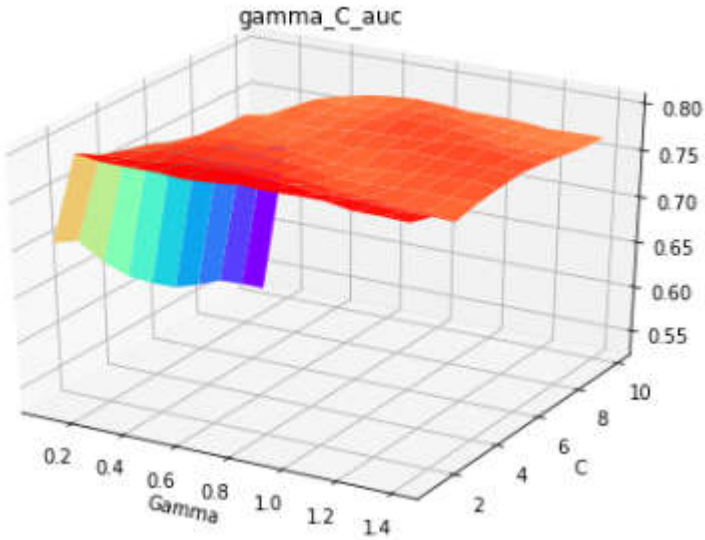
第一次实验：C=[1,15]，gamma=[1,11]，kernel=rbf，五次五折交叉验证：

```
最高准确率：  0.8057142857142857
```



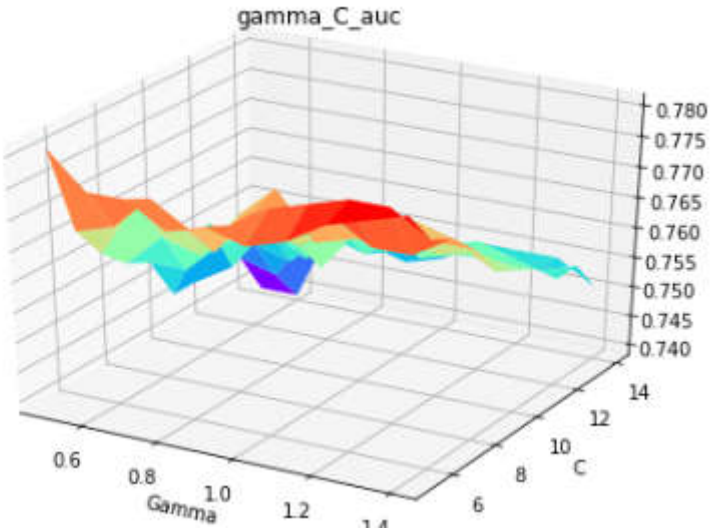
第二次实验：C=[1,15]，gamma=[1,11]，kernel=rbf，八次十折交叉验证：

```
最高准确率：  0.8041895604395604
```



第三次实验：C=[5,15]，gamma=[1,11]，kernel=rbf，五次五折交叉验证：

```
最高准确率：  0.7809523809523808
```



以上图片说明，三维模型的形状主要取决于给定的超参 C 与 gamma 范围。即 k-折交叉验证性能指标与超参数 C 与 gamma 取值范围密切相关。红颜色区域代表超参数最优值可能存在的范围。

参数最优值的选择与给定参数最优值范围，交叉验证参数 k 与 fold 有关。主要是和前者相关。此外 k 与 fold 对交叉验证性能指标有直接影响。

4.2.2 网格搜索算法

注：为了计算，输入输出结果将所有 C 值除以 10

第一次实验：C=[1,15]，gamma=[1,11]，kernel=rbf，五次五折交叉验证：

取 C=4，gamma=1 作为超参进行模型训练结果如下：

```
{'gamma': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'C': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4]}
The best parameters are {'C': 0.4, 'gamma': 1} with a score of 0.81
train: 0.8095238095238095
test: 0.7333333333333333
```

第二次实验：C=[1,15]，gamma=[1,11]，kernel=rbf，八次十折交叉验证：

取 C=2，gamma=1 作为超参进行模型训练结果如下：

```
{'gamma': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'C': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4]}
The best parameters are {'C': 0.2, 'gamma': 1} with a score of 0.80
train: 0.8285714285714286
test: 0.7555555555555555
```

第三次实验：C=[5,15]，gamma=[1,11]，kernel=rbf，五次五折交叉验证：

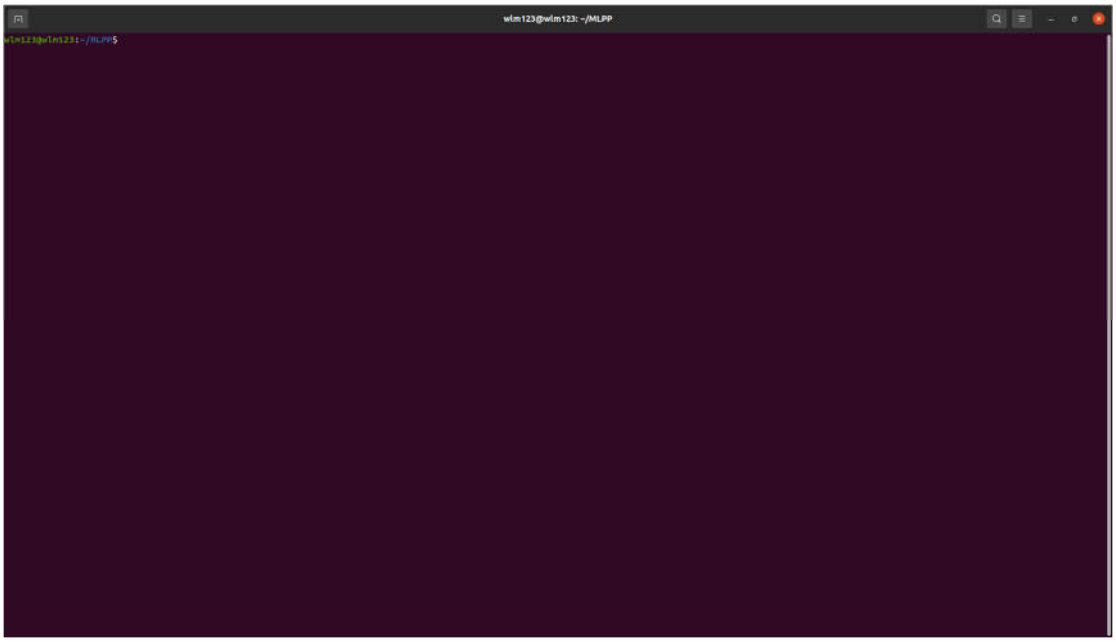
取 C=5，gamma=1 作为超参进行模型训练结果如下：

```
{'gamma': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'C': [0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4]}
The best parameters are {'C': 0.5, 'gamma': 1} with a score of 0.81
train: 0.8095238095238095
test: 0.7333333333333333
```

以上数据表明 k-折交叉验证参数 k 与 fold，与给定 C 和 gamma 范围共同限制了 C 与 gamma 的最优取值。该取值影响模型在测试集上的准确度。应该控制变量重复多做几次，确定实验范围内的最优超参数。本文最终最优超参数选为 C=2，gamma=1。

4.3 C++机器学习库 MLPP 的 SVC 实现

C++没有图形界面，运行结果将以 gif 显示



实验共进行 10 万次 epoch，图中列出了每次的权重和偏移量。最后显示了它的预测值及准确率是 91.036%。