# Machine Learning Lab5

Xiaoma

2023 年 1 月 20 日

## 1 实验要求

对于给定数据集 **train_feature.csv,train_label.csv,test_feature.csv**

- 数据预处理，进行数据降维、降噪、补缺、特征提取、编码以及必要的其他数据预处理工作
- 数据划分，将数据集拆分成训练集及测试集
- 模型训练，使用如下模型来完成标签预测任务
  - 线性回归模型
  - 决策树模型
  - 神经网络模型
  - 支持向量机
  - XGBoost
- 模型验证，对于 **test_feature.csv**，选择性能最佳的模型生成对应的数据标签并提交
- 撰写报告对以上任务进行相关分析

# 2 实验原理

## 2.1 数据预处理

### 2.1.1 特征选择

特征选择是特征工程里的一个重要问题，其目标是寻找最优特征子集。特征选择能剔除不相关 (irrelevant) 或冗余 (redundant ) 的特征，从而达到减少特征个数，提高模型精确度，减少运行时间的目的。另一方面，选取出真正相关的特征简化模型，协助理解数据产生的过程。然而在机器学习方面的成功很大程度上在于如果使用特征工程。

特征选择的方法

- Filter(过滤法)
- Wrapper(包装法)
- Embedded(嵌入法)

由于本次实现需要使用不同模型，故采用过滤法和嵌入法。

过滤法的基本思想是分别对每个特征 $x_i$ ，计算 $x_i$ 相对于类别标签 $y$ 的信息量 S(i) ，得到 n 个结果。然后将 n 个 S(i) 按照从大到小排序，输出前 k 个特征。

参考评价标准通常有

- Pearson 相关系数
- 卡方验证
- 距离相关系数
- 方差选择

嵌入法的基本思想是先使用某些机器学习的模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征。

### 2.1.2 降噪

数据中的随机错误或偏差被称为数据噪声，数据中较多的噪声会影响模型的鲁棒性。常见的噪声检测方法有：分箱、聚类、回归等。本次采用基于高斯分布进行噪声检测的方法。

首先利用极大似然估计法求高斯分布的参数设从正态总体 $N(\mu,\sigma^2)$ 抽出样本 $X_1,\Lambda,X_n$, 这里未知参数为 $mm\mu$ 和 $\sigma^2$ (注意我们把 $\sigma^2$ 看作一个参数)。似然函数为

$$\begin{aligned} L\left(\mu,\sigma^2\right) &= \prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x,-\mu)^2}{2\sigma^2}} \\ &= \left(2\pi\sigma^2\right)^{-\frac{n}{2}}e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2} \end{aligned}$$

它的对数为

$$\log L\left(\mu,\sigma^2\right) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\left(\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(x_i-\mu\right)^2,$$

似然方程组为

$$\begin{cases} \frac{\partial \log L\left(\mu,\quad \sigma^2\right)}{\partial\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left(x_i-\mu\right) = 0 \\ \frac{\partial \log L\left(\mu,\quad \sigma^2\right)}{\partial\sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}\left(x_i-\mu\right)^2 = 0, \end{cases}$$

由第一式解得

$$\mu^* = \bar{x} = \frac{1}{n}\sum_{i=1}^{n}x_i, \quad (7.2.3)$$

代入第二式得

$$\sigma^{*2} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2 \cdot (7.2.4)$$

似然方程组有唯一解 $(\mu^*,\sigma^{*2})$, 而且它一定是最大值点, 这是因为当 $|\mu|\to\infty$ 或 $\sigma^2\to 0$ 或 $\infty$ 时, 非负函数 $L(\mu,\sigma^2)\to 0$ 。于是 $\mu$ 和 $\sigma^2$ 的最大似然估计为

$$\mu^* = \bar{X}, \quad \sigma^{*2} = \frac{1}{n}\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2, (7.2.5)$$

这里, 我们用大写字母表示所有涉及的样本, 因为最大似然估计 $\mu^*$ 和 $\sigma^2$ 都是统计量, 离开了具体的一次试验或观测, 它们都是随机的。

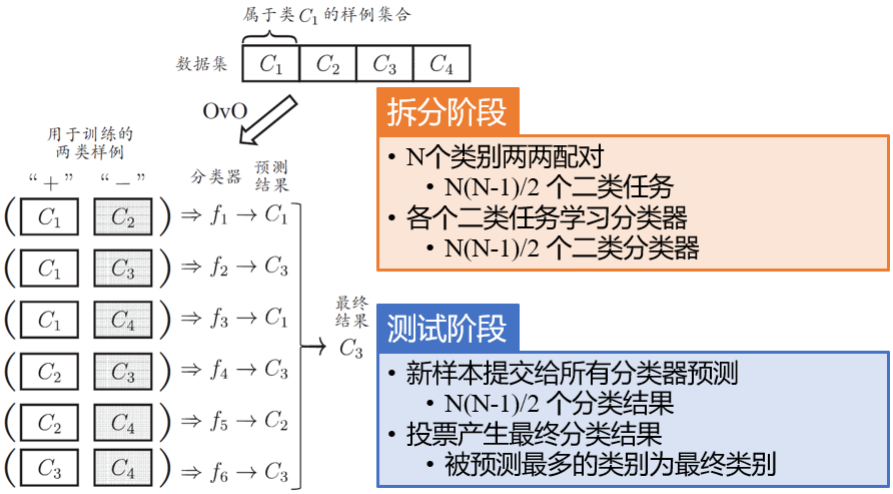
然后选择合适的阈值，在判断一个样本是否异常是，将该样本带入高斯函数计算概率，当概率小于阈值便判定这个样本出现异常。

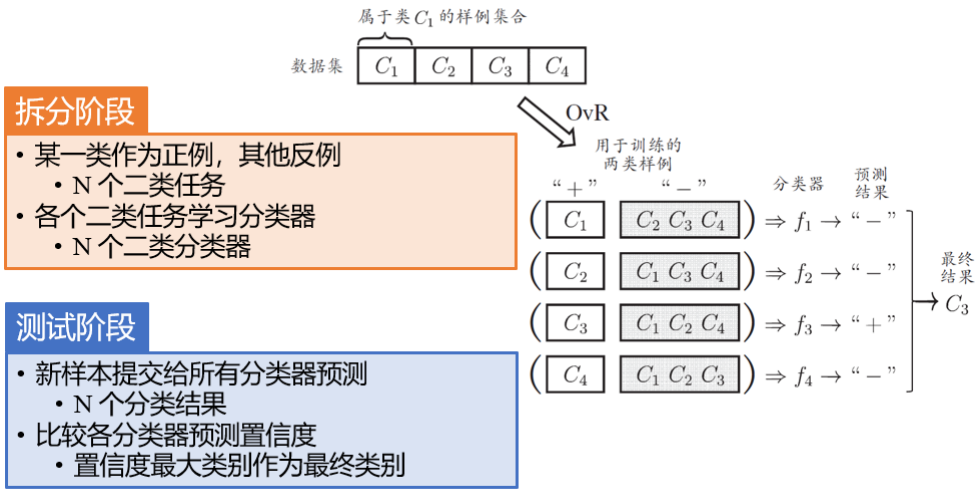
## 2.2 模型

### 2.2.1 线性多分类模型

将二分类模型推广到多分类，利用二分类学习器解决多分类问题。常见的方法有

- OVO ：对每个二分类任务训练一个分类器
- OVR ：对每个分类器的结果进行集成获得最终结果

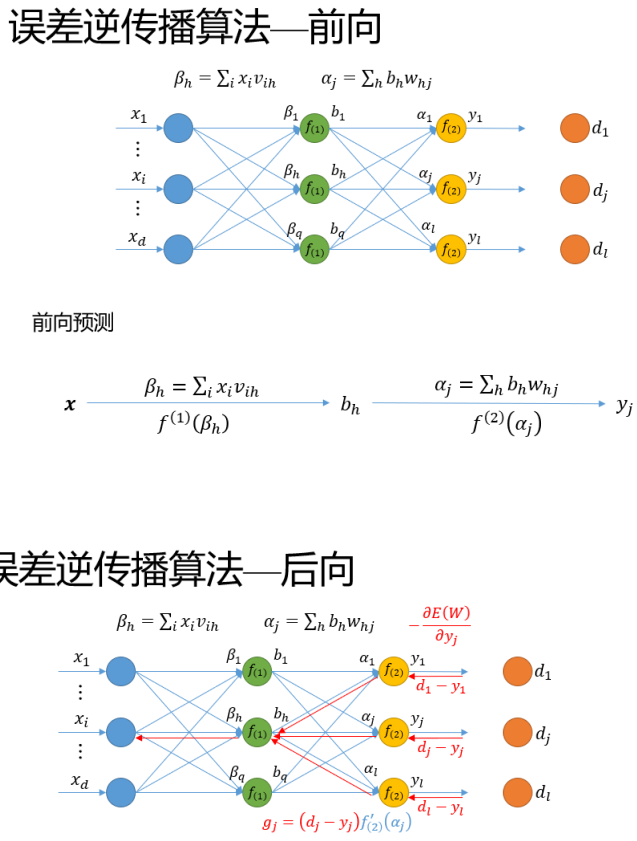


### 2.2.2 神经网络

本次实验中只使用了含有一个隐藏层的 MLP 模型，MLP 又称多层感知机，除了输入输出层以外，它中间可以含有多个隐藏层，最简单的 MLP 模型只有一个隐藏层。

多层感知机的层与层是全连接的，每个神经元都接受若干个输入并计算得到若干个输出传向下一层的神经元。通常我们通过误差逆传播（BP）算法来求解感知机中各项参数。



#### 2.2.3 决策树与 XGBoost

详见决策树与 XGBoost

#### 2.2.4 支持向量机

详见支持向量机

# 3 实验步骤

## 3.1 数据预处理

- 首先对查看数据集的信息，发现其存在缺失值，考虑到样本数量充足且有缺失值的样本量较少，故使用 **pandas.dropna 函数**将具有缺失值的样本舍弃。
- 对数据进行归一化,考虑到数据可能存在噪声,使用 **sklearn.covariance.EllipticEnvelop 函数**进行噪声检测，并将检测到的噪声样本舍弃。
- 考虑到使用整个数据集训练模型计算量过大且可能存在与分类任务相关性较低的特征，故对数据进行特征提取，分别使用 **shap 库**和 **XGboost.plot_importance 函数**选择权重最大的特征进行切片。

## 3.2 模型训练

首次训练未进行特征提取，所有模型的精确度都在 0.25 左右。

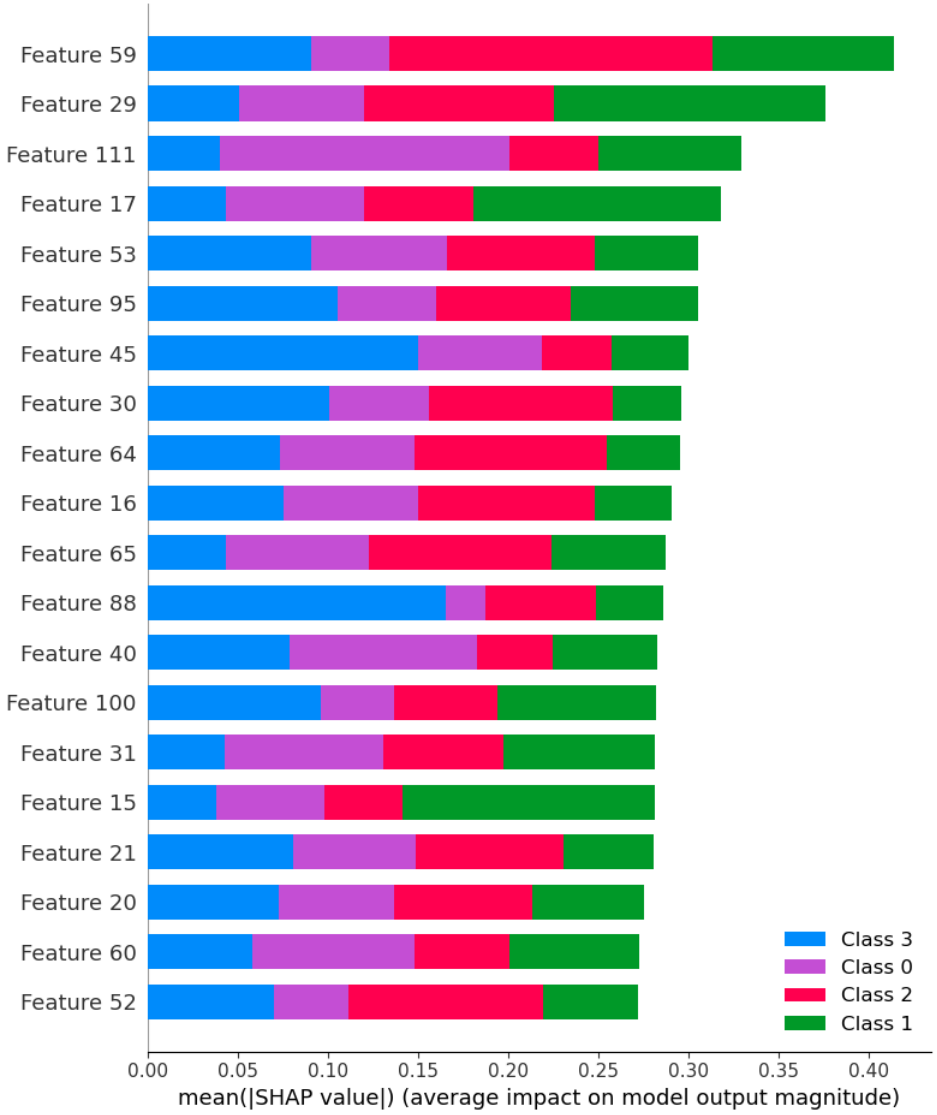
然后使用过滤法进行特征提取，分别使用卡方分布，F 分布，Pearson 相关系数等过滤特征，发现在卡方分布下，神经网络模型的精确度为 0.267，其他模型的精度仍在 0.25 左右。

```
1  neu_net_classfier = MLPClassifier(hidden_layer_sizes=(20,),
2                                    max_iter = 1000,
3                                    activation='relu',
4                                    solver = 'adam',
5                                    alpha = 0.0001,
6                                    )
```

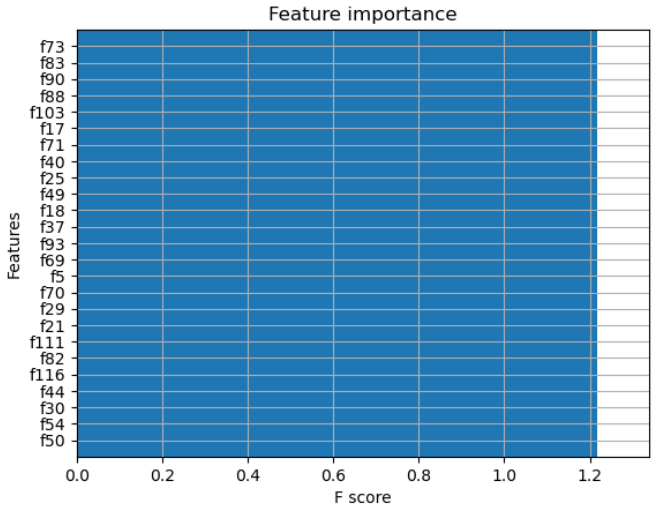
接下来进行嵌入法进行特征提取，使用基于 XGBoost 模型的特征提取，模型参数如下

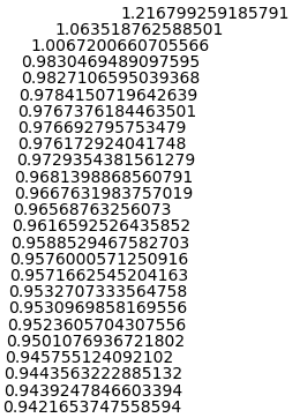
```
model = XGBClassifier(learning_rate = 1,
                      booster = 'gbtree',
                      max_depth = 20,
                      num_class = 4,
                      gamma = 0.1,
                      subsample = 1,
                      objective = 'muti:softprob',
                      eval_metric = 'mlogloss',
                      use_label_encoder = False,)
```

使用 **shap.summary_plot(shap_values, data, plot_type="bar")** 得到的条形图如下所示




使用 **plot_importance(importance_type='gain')** 得到的条形图如下所示



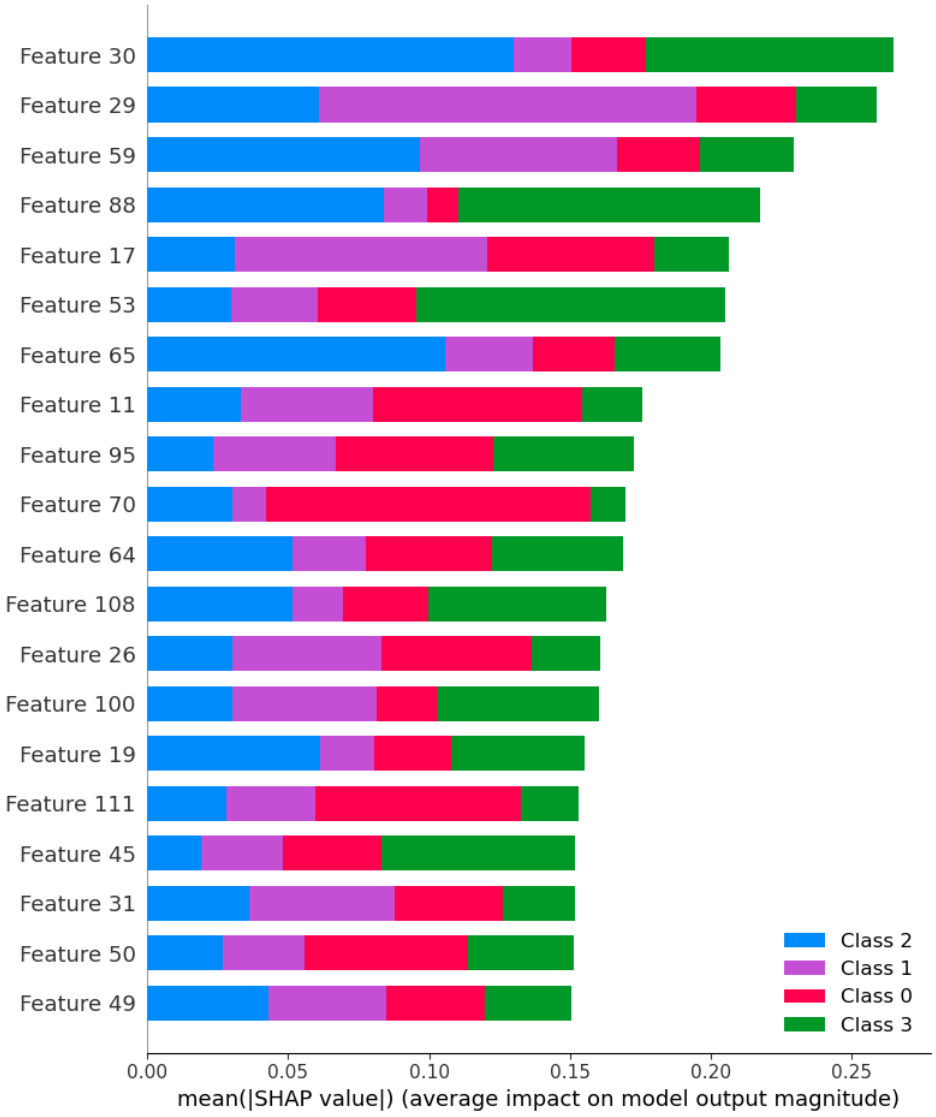




分别选择两图中权重较高的特征进行训练，得到的所有模型的精度仍在 0.25 左右。

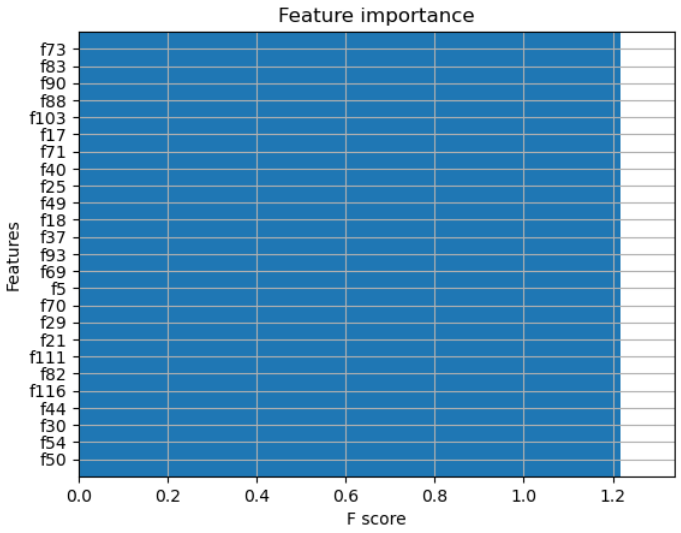
继续进行特征提取，更改参数

```
model = XGBClassifier(learning_rate = 0.1,
                      booster = 'gbtree',
                      max_depth = 20,
                      num_class = 4,
                      gamma = 0.1,
                      subsample = 1,
                      objective = 'muti:softprob',
                      eval_metric = 'mlogloss',
                      use_label_encoder = False,)
```

使用 **shap.summary_plot(shap_values, data, plot_type="bar")** 得到的条形图如下所示




使用 **plot_importance(importance_type='gain')** 得到的条形图如下所示



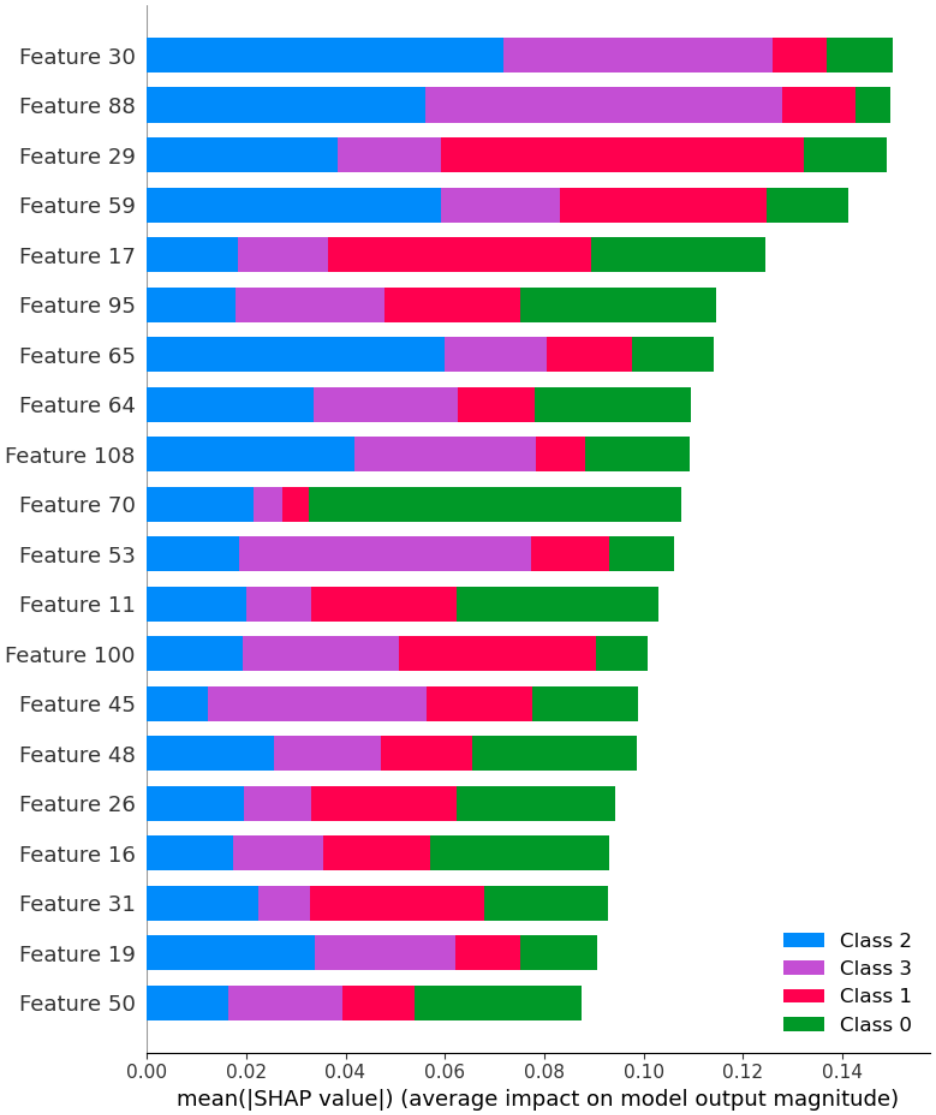

1.216799259185791
1.063518762588501
1.0067200660705566
0.9830469489097595
0.9827106595039368
0.9784150719642639
0.9767376184463501
0.976692795753479
0.976172924041748
0.9729354381561279
0.9681398868560791
0.9667631983757019
0.96568763256073
0.9616592526435852
0.9588529467582703
0.9576000571250916
0.9571662545204163
0.9532707333564758
0.9530969858169556
0.9523605704307556
0.9501076936721802
0.945755124092102
0.9443563222885132
0.9439247846603394
0.9421653747558594

分别选择两图中权重较高的特征进行训练，发现当选择特征为 [73, 90, 83, 88, 103, 17, 71, 40, 25] 时，XGBoost 模型的精确度达到 0.26，其他模型的精确度仍为 0.25 左右。

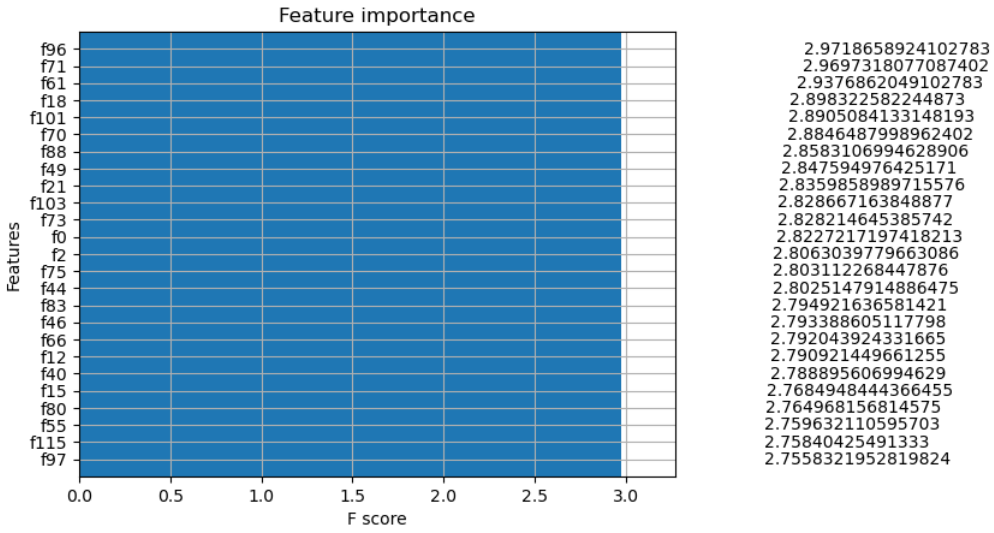
按照该方式不断调整，最终发现在该参数下

```
model = XGBClassifier(learning_rate = 0.1,
                      booster = 'gbtree',
                      max_depth = 20,
                      num_class = 4,
                      gamma = 2,
                      subsample = 1,
                      objective = 'muti:softprob',
                      eval_metric = 'mlogloss',
                      use_label_encoder = False,)
```

使用 **shap.summary_plot(shap_values, data, plot_type="bar")** 得到的条形图如下所示




使用 **plot_importance(importance_type='gain')** 得到的条形图如下所示




发现当选择特征为 30, 29, 59, 88, 17, 53, 65 时，XGBoost 模型的精确度达到 0.268，其他模型的精确度仍为 0.25 左右。

# 4 实验分析

实验所提供的数据集存在大量的无价值特征，可以通过特征选择的方法来提取有价值的特征进行模型训练以达到更好的结果，由于本人能力有限，无法提取出使模型表现非常好的特征组合。