机器学习导论

第一次作业

191220008 陈南曈

1 Basic Concepts

1.1 Probability

由贝叶斯公式：

$$\begin{aligned} Pr(D|T) &= \frac{Pr(T|D)Pr(D)}{Pr(T|D)Pr(D)+Pr(T|\neg D)Pr(\neg D)} \\ &= \frac{Pr(T|D)Pr(D)}{Pr(T|D)Pr(D)+Pr(T|\neg D)(1-Pr(D))} \\ &= \frac{0.98\times 0.01}{0.98\times 0.01+0.10\times(1-0.01)} \\ &= \frac{49}{544}=0.09007 \end{aligned}$$

故他真的得病概率为0.09007

1.2 Maximum likelihood estimation

似然函数为：

$$\ell(p)=C_{10}^{8}\cdot p^{8}(1-p)^{10-8}=45p^{8}(1-p)^{2}$$

对数似然为：

$$\ln \ell(p)=ln45+8lnp+2ln(1-p)$$

求导数：

$$(\ln \ell(p))'=\frac{8}{p}-\frac{2}{1-p}=\frac{8-10p}{p(1-p)}=0$$

得到 p 的估计值：

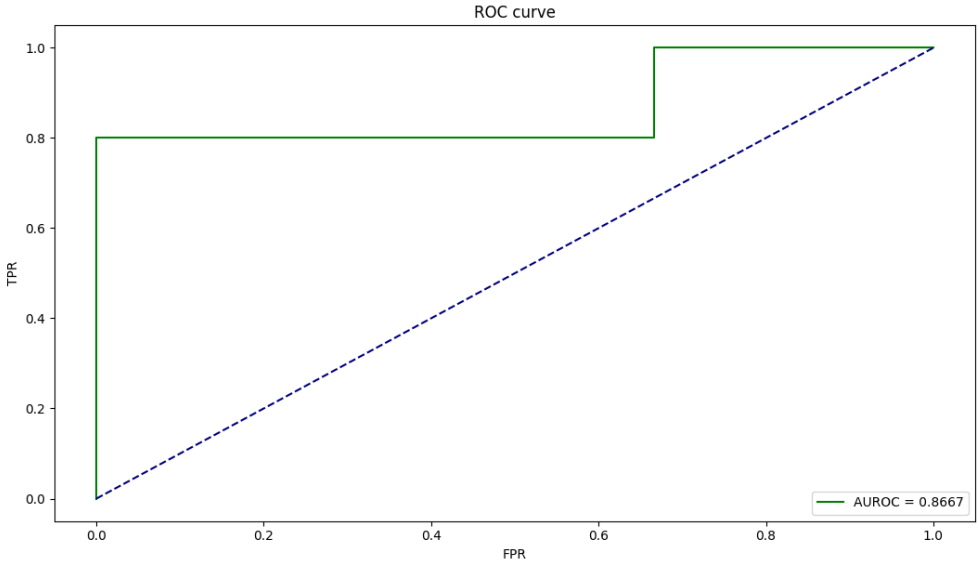
$$p=\frac{8}{10}=\frac{4}{5}=0.8$$

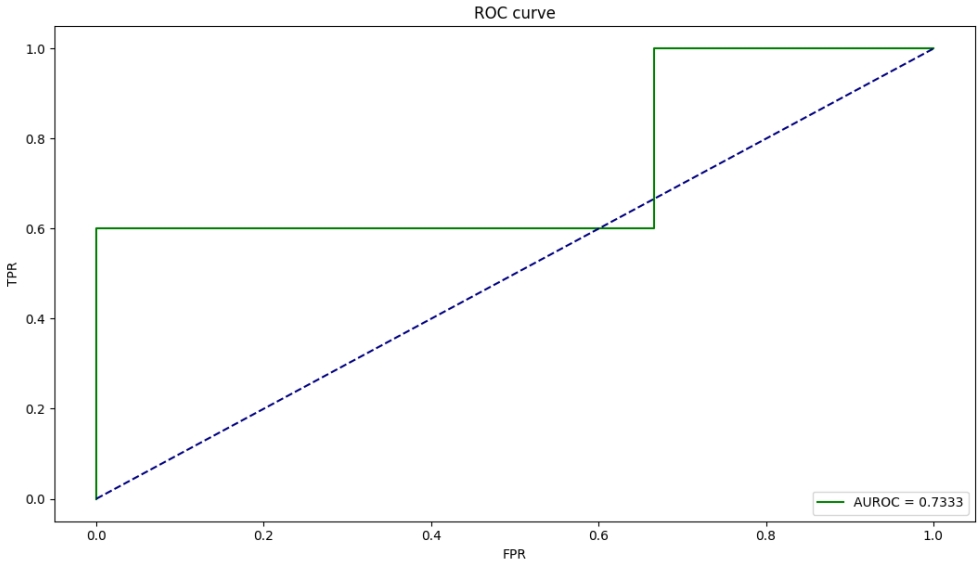
1.3 Performance meause

(1)

(实现代码见文件夹，直接运行即可)

在 python 中分别画出分类器 C1 和 C2 的 ROC 曲线，并计算 AUROC：





由图可知：

$$AUROC_1 = 0.8667$$
$$AUROC_2 = 0.7333$$

(2)

C1 (th1=0.40)		**预测结果**	**预测结果**
		正例	反例
真实结果	正例	4	1
真实结果	反例	0	3

$$F1 = \frac{2 \times TP}{样例总数 + TP - TN} = \frac{2 \times 4}{8 + 4 - 3} = \frac{8}{9} = 0.8889$$

C2 (th2=0.90)		预测结果	预测结果
		正例	反例
真实结果	正例	1	4
真实结果	反例	0	3

$$F1 = \frac{2 \times TP}{样例总数 + TP - TN} = \frac{2 \times 1}{8 + 1 - 3} = \frac{1}{3} = 0.3333$$

2 Linear model

1.

$$\begin{aligned} f(\mathbf{w}) &= \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \\ &= \frac{1}{2}||(\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\mathbf{w})\|_2 + \lambda||\mathbf{w}^{\mathbf{T}}\mathbf{w}||_2 \end{aligned}$$

对$\mathbf{w}$求偏导数：

$$\frac{\partial f}{\partial \mathbf{w}} = -\mathbf{X}^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\mathbf{w}) + 2\lambda\mathbf{w} = 0$$

由$\mathbf{X}$满秩可逆，得到$\mathbf{w}$的闭式解为：

$$\mathbf{w} = (2\lambda\mathbf{I} + \mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}$$

2.

由题可知：

$$\mathbf{X} = \begin{pmatrix} 2 & 9 & 1 \\ 9 & 3 & 1 \\ 8 & 3 & 1 \\ 8 & 8 & 1 \\ 2 & 1 & 1 \\ 8 & 4 & 1 \\ 4 & 3 & 1 \\ 1 & 8 & 1 \\ 3 & 3 & 1 \\ 5 & 3 & 1 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 290 \\ 1054 \\ 944 \\ 964 \\ 246 \\ 948 \\ 488 \\ 167 \\ 370 \\ 598 \end{pmatrix}$$

带入上述$\mathbf{w}$的闭式解，可得$\mathbf{w}$的最优解为：

$$\mathbf{w} = \begin{pmatrix} 112.93397617 \\ 6.18994302 \\ 11.97947962 \end{pmatrix}$$

3 Logistic Regression

1.

$$\ell(\beta)=\sum_{i=1}^{n}(-y_i\beta^{\mathbf{T}}\hat{\mathbf{x}}_i+\ln\left(1+e^{\beta^{\mathbf{T}}\hat{\mathbf{x}}_i}\right)) \tag{3.2}$$

欲证$\mathrm{Eq.}\ (3.2)$为凸函数，只需证明其 Hessian Matrix 半正定，即证：

$$\forall\mathbf{A}\in\mathbf{R}^m,\quad \mathbf{A}^{\mathbf{T}}\mathbf{H}\mathbf{A}\geq 0$$

由教材中公式(3.31)可知，关于β的二阶导数为：

$$\frac{\partial^2\ell(\beta)}{\partial\beta\partial\beta^{\mathbf{T}}}=\sum_{i=1}^{n}\hat{\mathbf{x}}_i\hat{\mathbf{x}}_i^{\mathbf{T}}p_1(\hat{\mathbf{x}}_i;\beta)(1-p_1(\hat{\mathbf{x}}_i;\beta))$$

设$\beta\in\mathbf{R}^m,\hat{\mathbf{x}}\in\mathbf{R}^m,p_1=p_1(\hat{\mathbf{x}}_i;\beta)$，则 Hessian Matrix 为：

$$\mathbf{H}_{m\times m}=\sum_{i=1}^{n}\hat{\mathbf{x}}_i\hat{\mathbf{x}}_i^{\mathbf{T}}p_1(1-p_1)=\mathbf{X}\mathbf{P}\mathbf{X}^{\mathbf{T}}$$

其中：

$$\mathbf{X}=[\hat{\mathbf{x}}_1,\ \hat{\mathbf{x}}_2,\ \cdots,\ \hat{\mathbf{x}}_n]_{m\times n}$$

$$\mathbf{P}=\begin{bmatrix}p_1(1-p_1) & 0 & \cdots & 0\\ 0 & p_2(1-p_2) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & p_n(1-p_n)\end{bmatrix}_{n\times n}$$

由于：

$$p_i=\frac{1}{1+e^{\beta_i^{\mathbf{T}}\hat{\mathbf{x}}_i}}\in[0,1]$$

可得：

$$p_i(1-p_i)\geq 0$$

故对$\forall\mathbf{A}\in\mathbf{R}^m$有：

$$\begin{aligned}&\mathbf{A}^{\mathbf{T}}\mathbf{H}\mathbf{A}\\ =\ &\mathbf{A}^{\mathbf{T}}\mathbf{X}\mathbf{P}\mathbf{X}^{\mathbf{T}}\mathbf{A}\\ =\ &(\mathbf{X}^{\mathbf{T}}\mathbf{A})^{\mathbf{T}}\mathbf{P}(\mathbf{X}^{\mathbf{T}}\mathbf{A})\\ \overset{\mathbf{B}=\mathbf{X}^{\mathbf{T}}\mathbf{A}}{=}\ &\mathbf{B}^{\mathbf{T}}\mathbf{P}\mathbf{B}\\ =\ &\sum_{i=1}^{n}(B_i)^2p_i(1-p_i)\geq 0\end{aligned}$$

因此，Hessian Matrix 为半正定矩阵，因而式(3.2)为凸函数，证毕。

2.

若该二分类问题变为多分类问题，其中$y_i \in \{1, 2, \cdots, K\}$，可看作求解K个二分类问题。

假设$y_i = K$为主类别，则其余$K-1$个类别满足对数几率：

$$\begin{aligned}
\ln \frac{p(y=1 \mid \mathbf{x})}{p(y=K \mid \mathbf{x})} &= \mathbf{w}_1^{\mathrm{T}} \mathbf{x} + b_1 \\
\ln \frac{p(y=2 \mid \mathbf{x})}{p(y=K \mid \mathbf{x})} &= \mathbf{w}_2^{\mathrm{T}} \mathbf{x} + b_2 \\
&\cdots \\
\ln \frac{p(y=K-1 \mid \mathbf{x})}{p(y=K \mid \mathbf{x})} &= \mathbf{w}_{K-1}^{\mathrm{T}} \mathbf{x} + b_{K-1}
\end{aligned}$$

因此有：

$$\frac{p(y=i \mid \mathbf{x})}{p(y=K \mid \mathbf{x})} = e^{\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i}, \quad (i = 1, 2, \ldots, K-1)$$

求和得：

$$\sum_{i=1}^{K-1} \frac{p(y=i \mid \mathbf{x})}{p(y=K \mid \mathbf{x})} = \sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i} = \frac{1 - p(y=K \mid \mathbf{x})}{p(y=K \mid \mathbf{x})}$$

解得：

$$p(y=j \mid \mathbf{x}) = \begin{cases} \frac{e^{\mathbf{w}_j^{\mathrm{T}} \mathbf{x} + b_j}}{1 + \sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i}}, & j = 1, 2, \ldots, K-1 \\ \\ \frac{1}{1 + \sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i}}, & j = K \end{cases}$$

令$\beta_i = (\mathbf{w}_i; b_i)$，$\hat{\mathbf{x}}_i = (\mathbf{x}_i; 1)$，则对数似然为：

$$
\begin{aligned}
\ell(\beta) &= \sum_{i=1}^{n} \ln p(y_i \mid \hat{x}_i) \\
&= \sum_{i=1}^{n} \ln \prod_{j=1}^{K} (p(y_i = j \mid \hat{x}_i))^{\mathbb{I}(y_i = j)} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{K} \mathbb{I}(y_i = j) \ln p(y_i = j \mid \hat{x}_i) \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \ln p(y_i = j \mid \hat{x}_i) + \mathbb{I}(y_i = K) \ln p(y_i = K \mid \hat{x}_i)) \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \ln \left(\frac{e^{\mathbf{w}_j^{\mathrm{T}} \mathbf{x}_i + b_j}}{1 + \sum_{m=1}^{K-1} e^{\mathbf{w}_m^{\mathrm{T}} \mathbf{x}_i + b_m}} \right) + \mathbb{I}(y_i = K) \ln \left(\frac{1}{1 + \sum_{m=1}^{K-1} e^{\mathbf{w}_m^{\mathrm{T}} \mathbf{x}_i + b_m}} \right)) \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \ln \left(\frac{e^{\beta_j^{\mathbf{T}} \hat{\mathbf{x}}_i}}{1 + \sum_{m=1}^{K-1} e^{\beta_m^{\mathbf{T}} \hat{\mathbf{x}}_i}} \right) + \mathbb{I}(y_i = K) \ln \left(\frac{1}{1 + \sum_{m=1}^{K-1} e^{\beta_m^{\mathbf{T}} \hat{\mathbf{x}}_i}} \right)) \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) (\beta_j^{\mathbf{T}} \hat{\mathbf{x}}_i - \ln \left(1 + \sum_{m=1}^{K-1} e^{\beta_m^{\mathbf{T}} \hat{\mathbf{x}}_i} \right)) - \mathbb{I}(y_i = K) \ln \left(1 + \sum_{m=1}^{K-1} e^{\beta_m^{\mathbf{T}} \hat{\mathbf{x}}_i} \right)) \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \beta_j^{\mathbf{T}} \hat{\mathbf{x}}_i - \sum_{j=1}^{K} \mathbb{I}(y_i = j) \ln \left(1 + \sum_{m=1}^{K-1} e^{\beta_m^{\mathbf{T}} \hat{\mathbf{x}}_i} \right)) \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \beta_j^{\mathbf{T}} \hat{\mathbf{x}}_i - \ln \left(1 + \sum_{m=1}^{K-1} e^{\beta_m^{\mathbf{T}} \hat{\mathbf{x}}_i} \right))
\end{aligned}
$$

3.

（实现代码见文件夹，直接运行即可）

读取 Yeast 数据集后可知是多分类数据集，故可以采用不同的拆分方式进行训练，

如：OVO，OVR，Multi-Class

我将数据集的70%用于训练，30%用于测试（同时也会在训练集上进行测试）。

此外，我在数据集划分时让各类别在训练集和测试集中的比例分布相同，并且保证每一次的划分始终保持相同，以确保测试的科学性。

在模型评估中，我选取了精度、查准率、查全率、f1、kappa系数、混淆矩阵等参数进行评估，并分别计算出其宏平均和微平均，通过分类报告列举出上述部分参数在各类别中的值。

测试集：

OVO:

```
accuracy: 58.97%
precision(宏平均): 60.31%, precision(微平均): 58.97%
recall(宏平均): 55.25%, recall(微平均): 58.97%
f1(宏平均): 57.16%, f1(微平均): 58.97%
cohen kappa:  0.4617682552871585
```

```
confusion matrix:
 [[96  0  0  0  0  1  6 34  1  1]
 [ 0  2  0  0  0  0  0  0  0  0]
 [ 1  0  6  2  0  0  2  0  0  0]
 [ 0  0  1  8  4  0  0  0  0  0]
 [ 3  0  1  1  5  3  0  2  0  0]
 [ 1  0  0  0  1 39  1  7  0  0]
 [21  0  0  0  1  4 40  6  1  0]
 [54  0  0  0  0  5  6 64  0  0]
 [ 2  0  0  0  0  0  1  0  3  0]
 [ 4  0  0  0  0  4  0  1  0  0]]
```

```
classification report:
              precision    recall  f1-score   support

           0       0.53      0.69      0.60       139
           1       1.00      1.00      1.00         2
           2       0.75      0.55      0.63        11
           3       0.73      0.62      0.67        13
           4       0.45      0.33      0.38        15
           5       0.70      0.80      0.74        49
           6       0.71      0.55      0.62        73
           7       0.56      0.50      0.53       129
           8       0.60      0.50      0.55         6
           9       0.00      0.00      0.00         9

    accuracy                           0.59       446
   macro avg       0.60      0.55      0.57       446
weighted avg       0.59      0.59      0.58       446
```

OVR:

```
accuracy: 58.30%
precision(宏平均): 59.40%, precision(微平均): 58.30%
recall(宏平均): 51.08%, recall(微平均): 58.30%
f1(宏平均): 53.43%, f1(微平均): 58.30%
cohen kappa:  0.44957999920379
```

```
confusion matrix:
 [[96  0  0  0  0  1  6 35  1  0]
 [ 0  2  0  0  0  0  0  0  0  0]
 [ 3  0  3  2  0  0  3  0  0  0]
 [ 1  0  1  6  3  0  2  0  0  0]
 [ 3  0  0  1  5  2  1  3  0  0]
 [ 3  0  0  0  0 39  1  6  0  0]
 [21  0  0  0  1  4 39  6  2  0]
 [52  0  0  0  0  5  5 67  0  0]
 [ 2  0  0  0  0  0  1  0  3  0]
 [ 4  0  0  0  0  4  0  1  0  0]]
```

```
classification report:
              precision    recall  f1-score   support

           0       0.52      0.69      0.59       139
           1       1.00      1.00      1.00         2
           2       0.75      0.27      0.40        11
           3       0.67      0.46      0.55        13
           4       0.56      0.33      0.42        15
           5       0.71      0.80      0.75        49
           6       0.67      0.53      0.60        73
           7       0.57      0.52      0.54       129
           8       0.50      0.50      0.50         6
           9       0.00      0.00      0.00         9

    accuracy                           0.58       446
   macro avg       0.59      0.51      0.53       446
weighted avg       0.58      0.58      0.57       446
```

Multi-Class：

```
accuracy: 57.85%
precision(宏平均): 58.96%, precision(微平均): 57.85%
recall(宏平均): 48.36%, recall(微平均): 57.85%
f1(宏平均): 51.83%, f1(微平均): 57.85%
cohen kappa:  0.4457979444132325
```

```
confusion matrix:
 [[96  0  0  0  0  0  6 35  1  1]
 [ 1  1  0  0  0  0  0  0  0  0]
 [ 1  0  5  2  0  0  3  0  0  0]
 [ 0  0  1  7  4  0  1  0  0  0]
 [ 3  0  0  1  5  3  1  2  0  0]
 [ 2  0  0  0  0 40  1  6  0  0]
 [23  0  0  0  2  4 37  5  2  0]
 [54  0  0  0  0  5  6 64  0  0]
 [ 2  0  0  1  0  0  0  0  3  0]
 [ 5  0  0  0  0  4  0  0  0  0]]
```

```
classification report:
              precision    recall  f1-score   support

           0       0.51      0.69      0.59       139
           1       1.00      0.50      0.67         2
           2       0.83      0.45      0.59        11
           3       0.64      0.54      0.58        13
           4       0.45      0.33      0.38        15
           5       0.71      0.82      0.76        49
           6       0.67      0.51      0.58        73
           7       0.57      0.50      0.53       129
           8       0.50      0.50      0.50         6
           9       0.00      0.00      0.00         9

    accuracy                           0.58       446
   macro avg       0.59      0.48      0.52       446
weighted avg       0.58      0.58      0.57       446
```

由上图可以得出，该模型在测试集上的测试结果为：

在所有的参数上（精度、查准率、查全率、f1、kappa系数），均有：OVO > OVR > Multi-Class

但我发现，上述比较关系的结果与数据集划分的随机种子的值有关，三者的相对大小并不固定。

此外，参数的值（如精度）均不太高，cohen kappa 均小于 0.8，说明都不是好的分类。

部分的类别，如类别9（ERL），甚至没有被预测到，导致其相关参数均为0，说明学习效果并不理想。

训练集：

OVO：

```
accuracy: 62.04%
precision(宏平均): 66.13%, precision(微平均): 62.04%
recall(宏平均): 60.81%, recall(微平均): 62.04%
f1(宏平均): 62.59%, f1(微平均): 62.04%
cohen kappa:  0.503675359617282
```

```
confusion matrix:
 [[231   0   0   0   3   3  29  58   0   0]
 [  0   3   0   0   0   0   0   0   0   0]
 [  3   0  15   0   2   0   3   1   0   0]
 [  0   0   1  26   3   1   0   0   0   0]
 [  4   0   0   4  17   4   6   1   0   0]
 [  6   0   0   0   1  99   2   6   0   0]
 [ 55   0   0   1   4   3  97  10   1   0]
 [126   0   1   0   1   7  16 149   0   0]
 [  3   0   0   0   1   1   2   0   7   0]
 [  9   0   2   1   1   2   3   3   0   0]]
```

```
classification report:
              precision    recall  f1-score   support

           0       0.53      0.71      0.61       324
           1       1.00      1.00      1.00         3
           2       0.79      0.62      0.70        24
           3       0.81      0.84      0.83        31
           4       0.52      0.47      0.49        36
           5       0.82      0.87      0.85       114
           6       0.61      0.57      0.59       171
           7       0.65      0.50      0.56       300
           8       0.88      0.50      0.64        14
           9       0.00      0.00      0.00        21

    accuracy                           0.62      1038
   macro avg       0.66      0.61      0.63      1038
weighted avg       0.62      0.62      0.61      1038
```

OVR：

```
accuracy: 60.50%
precision(宏平均): 64.44%, precision(微平均): 60.50%
recall(宏平均): 55.59%, recall(微平均): 60.50%
f1(宏平均): 58.50%, f1(微平均): 60.50%
cohen kappa:  0.480362493498121
```

```
confusion matrix:
 [[230   0   0   1   0   2  28  63   0   0]
 [  0   3   0   0   0   0   0   0   0   0]
 [  6   0  10   1   2   0   4   1   0   0]
 [  0   0   2  21   2   2   4   0   0   0]
 [  6   0   0   5  12   3   9   1   0   0]
 [  9   0   0   0   0  98   0   7   0   0]
 [ 55   0   0   1   3   4  95  12   1   0]
 [122   0   1   0   0  11  14 152   0   0]
 [  5   0   0   0   0   0   2   0   7   0]
 [ 10   0   2   0   0   4   2   3   0   0]]
```

```
classification report:
              precision    recall  f1-score   support

           0       0.52      0.71      0.60       324
           1       1.00      1.00      1.00         3
           2       0.67      0.42      0.51        24
           3       0.72      0.68      0.70        31
           4       0.63      0.33      0.44        36
           5       0.79      0.86      0.82       114
           6       0.60      0.56      0.58       171
           7       0.64      0.51      0.56       300
           8       0.88      0.50      0.64        14
           9       0.00      0.00      0.00        21

    accuracy                           0.61      1038
   macro avg       0.64      0.56      0.59      1038
weighted avg       0.61      0.61      0.60      1038
```

Multi-Class:

```
accuracy: 61.75%
precision(宏平均): 64.68%, precision(微平均): 61.75%
recall(宏平均): 59.09%, recall(微平均): 61.75%
f1(宏平均): 60.96%, f1(微平均): 61.75%
cohen kappa:  0.4994807589300354
```

```
confusion matrix:
 [[233   0   0   0   2   3  29  57   0   0]
 [  0   3   0   0   0   0   0   0   0   0]
 [  3   0  13   2   3   0   2   1   0   0]
 [  0   0   2  24   3   1   1   0   0   0]
 [  5   0   0   4  16   3   7   1   0   0]
 [  7   0   0   0   1  99   0   7   0   0]
 [ 55   0   0   2   5   3  96   9   1   0]
 [124   0   1   0   1   8  16 150   0   0]
 [  4   0   0   0   1   0   2   0   7   0]
 [  9   0   2   0   0   4   3   3   0   0]]
```

```
classification report:
              precision    recall  f1-score   support

           0       0.53      0.72      0.61       324
           1       1.00      1.00      1.00         3
           2       0.72      0.54      0.62        24
           3       0.75      0.77      0.76        31
           4       0.50      0.44      0.47        36
           5       0.82      0.87      0.84       114
           6       0.62      0.56      0.59       171
           7       0.66      0.50      0.57       300
           8       0.88      0.50      0.64        14
           9       0.00      0.00      0.00        21

    accuracy                           0.62      1038
   macro avg       0.65      0.59      0.61      1038
weighted avg       0.62      0.62      0.61      1038
```

由上图可以得出，该模型在训练集上的测试结果为：

在所有的参数上（精度、查准率、查全率、f1、kappa系数），均有：OVO > Multi-Class > OVR

但我发现，上述比较关系的结果与数据集划分的随机种子的值有关，三者的相对大小并不固定。

此外，参数的值（如精度）均不太高，cohen kappa 均小于 0.8，说明都不是好的分类。

部分的类别，如类别9（ERL），甚至没有被预测到，导致其相关参数均为0，说明学习效果并不理想。