## SVM实例

## 运行结果

Fitting 5 folds for each of 6 candidates, totalling 30 fits {'C': 5} 0.698

precision	recall	f1-score	support	
0.85	0.87	0.86	207	
0.63	0.72	0.67	190	
0.66	0.81	0.73	187	
0.71	0.78	0.74	186	
0.67	0.48	0.56	192	
0.81	0.71	0.75	194	
0.53	0.26	0.35	184	
0.43	0.40	0.42	181	
0.87	0.86	0.86	204	
0.89	0.78	0.83	192	
0.66	0.57	0.61	192	
0.76	0.81	0.79	179	
0.80	0.91	0.85	196	
0.74	0.74	0.74	176	
0.58	0.40	0.47	173	
0.81	0.84	0.82	201	
0.61	0.78	0.68	195	
0.62	0.79	0.69	189	
0.47	0.46	0.47	178	
0.82	0.89	0.85	211	
0.85	0.82	0.83	202	
0.80	0.77	0.79	190	
0.87	0.86	0.86	198	
0.66	0.75	0.70	207	
0.74	0.73	0.74	211	
0.73	0.77	0.75	185	
		0.72	5000	
0.71	0.71	0.71	5000	
0.72	0.72	0.71	5000	
	0.85 0.63 0.66 0.71 0.67 0.81 0.53 0.43 0.87 0.89 0.66 0.76 0.80 0.74 0.58 0.81 0.61 0.62 0.47 0.82 0.85 0.80 0.74 0.73	0.85       0.87         0.63       0.72         0.66       0.81         0.71       0.78         0.67       0.48         0.81       0.71         0.53       0.26         0.43       0.40         0.87       0.86         0.89       0.78         0.66       0.57         0.76       0.81         0.80       0.91         0.74       0.74         0.58       0.40         0.81       0.84         0.61       0.78         0.62       0.79         0.47       0.46         0.82       0.89         0.85       0.82         0.80       0.77         0.87       0.86         0.66       0.75         0.74       0.73         0.73       0.77	0.85       0.87       0.86         0.63       0.72       0.67         0.66       0.81       0.73         0.71       0.78       0.74         0.67       0.48       0.56         0.81       0.71       0.75         0.53       0.26       0.35         0.43       0.40       0.42         0.87       0.86       0.86         0.89       0.78       0.83         0.66       0.57       0.61         0.76       0.81       0.79         0.80       0.91       0.85         0.74       0.74       0.74         0.58       0.40       0.47         0.81       0.84       0.82         0.62       0.79       0.69         0.47       0.46       0.47         0.82       0.89       0.85         0.85       0.82       0.83         0.80       0.77       0.79         0.87       0.86       0.86         0.66       0.75       0.70         0.74       0.73       0.74         0.73       0.74       0.75         0.72       0.71	0.85       0.87       0.86       207         0.63       0.72       0.67       190         0.66       0.81       0.73       187         0.71       0.78       0.74       186         0.67       0.48       0.56       192         0.81       0.71       0.75       194         0.53       0.26       0.35       184         0.43       0.40       0.42       181         0.87       0.86       0.86       204         0.89       0.78       0.83       192         0.66       0.57       0.61       192         0.76       0.81       0.79       179         0.80       0.91       0.85       196         0.74       0.74       0.74       176         0.58       0.40       0.47       173         0.81       0.84       0.82       201         0.61       0.78       0.68       195         0.62       0.79       0.69       189         0.47       0.46       0.47       178         0.82       0.89       0.85       211         0.85       0.82       0.83

## 代码分析

这段代码用于支持向量机(SVM)分类任务的建模和评估。

```
predictors = letters.columns[1:]
X_train, X_test, y_train, y_test =
model_selection.train_test_split(letters[predictors], letters.letter,
test_size=0.25, random_state=1234)
```

将数据集拆分为输入特征X和目标变量y,其中X包含除'letter'列之外的所有列,y是'letter'列。然后,使用train\_test\_split函数将数据集划分为训练集和测试集,其中测试集占比为0.25。

```
C = [0.05, 0.1, 0.5, 1, 2, 5]
parameters = {'C': C}
grid_linear_svc = model_selection.GridSearchCV(estimator=svm.LinearSVC(
    dual=False), param_grid=parameters, scoring='accuracy', cv=5, verbose=1)
```

使用GridSearchCV函数进行网格搜索,以选择线性可分SVM模型中最佳的C值(正则化参数)。参数C的候选值为[0.05, 0.1, 0.5, 1, 2, 5]。使用accuracy作为评分指标进行交叉验证(cv=5),同时设置verbose参数为1以显示搜索过程的详细信息。

```
grid_linear_svc.fit(X_train, y_train)
```

对网格搜索对象grid\_linear\_svc进行训练,通过拟合训练数据集X\_train和y\_train来搜索最佳的C值。

```
grid_linear_svc.best_params_, grid_linear_svc.best_score_
```

返回交叉验证后得到的最佳参数值和相应的评分。

```
pred_linear_svc = grid_linear_svc.predict(X_test)
```

使用训练好的模型对测试集X\_test进行预测,将预测结果存储在pred\_linear\_svc中。

```
report = metrics.classification_report(y_test, pred_linear_svc)
print(report)
```

这段代码使用classification\_report函数计算模型的预测准确率,并将结果打印出来。