機器學習 HW3

2023/05/11

611121213 莊雅卉

1. LogisticRegression
2. LogisticRegression (C=1e5,class\_weight='balanced')

**Accuracy : 97.0%**

1. LogisticRegression (C=1e5,penalty='l2')

Accuracy : 94.2%

1. LogisticRegression (C=1e5)

Accuracy : 96.4%

1. LogisticRegression (C=1e5,penalty='l2',multi\_class='multinomial')

Accuracy : 94.1%

1. LogisticRegression (C=1e5,penalty='l1',solver='liblinear')

Accuracy : 92.3%

1. LogisticRegression (C=1e5,class\_weight='balanced',solver='liblinear')

Accuracy : 95.3%

1. LogisticRegression (C=1e5,penalty='l2',solver='liblinear')

Accuracy : 95.9%

1. LogisticRegression (C=1e5,penalty='l2',solver='saga')

Accuracy : 90.6%

1. LogisticRegression (C=1e5,penalty='l1',solver='saga')

Accuracy : 90.0%

1. RandomForestClassifier
2. RandomForestClassifier(n\_estimators=100)

Train\_Accuracy:100%,Test\_Accuracy : 95.3%

1. RandomForestClassifier (n\_estimators=100,criterion='gini')

Train\_Accuracy:100.0%,Test\_Accuracy : 96.4%

1. RandomForestClassifier (n\_estimators=100,criterion='entropy')

Train\_Accuracy:100.0%,Test\_Accuracy : 98.2%

1. RandomForestClassifier(n\_estimators=100,criterion='entropy',max\_depth=5)

Train\_Accuracy:98.9%,Test\_Accuracy : 95.9%

1. RandomForestClassifier(n\_estimators=100,criterion='gini',max\_depth=5 )

Train\_Accuracy:99.7%,Test\_Accuracy : 92.9%

1. RandomForestClassifier(n\_estimators=100,class\_weight='balanced' )

Train\_Accuracy:100.0%,Test\_Accuracy : 96.4%

1. RandomForestClassifier(n\_estimators=100, class\_weight='balanced\_subsample')

Train\_Accuracy:100.0%,**Test\_Accuracy : 98.2%**

1. SVC
2. SVC(C=1e5,kernel="linear",gamma="scale",class\_weight="balanced")

Train\_Accuracy:100.0%,**Test\_Accuracy : 98.2%**

1. SVC(C=1e5,kernel="linear",gamma="scale")

Train\_Accuracy:97.2%,Test\_Accuracy : 94.7%

1. SVC(C=1e5,kernel="linear",gamma="auto",class\_weight="balanced")

Train\_Accuracy:98.2%,Test\_Accuracy : 92.3%

1. SVC(C=1e5,kernel="linear",gamma="auto")

Train\_Accuracy:98.2%,Test\_Accuracy : 94.1%

1. SVC(C=1e5,kernel="poly",gamma="auto",class\_weight="balanced")

Train\_Accuracy:98.2%,Test\_Accuracy : 94.1%

1. BaggingClassifier
2. BaggingClassifier(base\_estimator=SVC(), n\_estimators=10)

Accuracy : 91.8%

1. BaggingClassifier(base\_estimator=SVC(C=1e5,kernel="linear",gamma="scale",class\_weight="balanced"), n\_estimators=10)

**Accuracy : 96.4%**

1. VotingClassifier
2. VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('gnb', clf3)], voting='hard')

Accuracy : 94.1%

1. VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('gnb', clf3)])

**Accuracy : 97.1%**

1. VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('gnb', clf3)], voting='hard', n\_jobs=12)

Accuracy : 96.4%

1. 統整結果
2. LogisticRegression = 97.0%
3. RandomForestClassifier = 98.2%
4. SVC = 98.2%
5. BaggingClassifier= 96.4%
6. VotingClassifier = 97.1%
7. 結論

每種方法透過調整參數後皆可以提升準確度，在LogisticRegression情境中，參數penalty='l2',multi\_class='multinomial'為本實驗中準確率最好的一個配置；而在RandomForestClassifier情境中無論是使用gini還是entropy皆能達到很好的分類，而max\_depth設定的高反而會使準確度下降；在SVC情境中，參數設定為C=1,kernel='rbf'已經能夠達到82%的準確率，但再新增設置gamma=0.7時，能夠再將準確率提升一些些，而在kernel='sigmoid'時，準確度明顯地低於rbf、poly；在KNeighbors-Classifier 情境中，最主要的參數為二，一個是weights 在設定為uniform時的表現不如distance，而n\_neighbors在5時的表現度優於3；最後是GaussianNB、MultinomialNB，在本次實驗中GaussianNB的訓練結果優於MultinomialNB，調整了其中的參數也不見好轉。

就這五種分類法而言，RandomForestClassifier、KNeighborsClassifier表現為最佳，GaussianNB MultinomialNB為最差。準確度會依照資料集、參數、方法等因素會有不同的表現，無法斬釘截鐵地說哪個方法是最佳的，而是要依照資料集的屬性去做適合他的分類方式。

透過本次作業能夠清楚的了解五種分類法的參數設定，以及圖片的樣貌，來對機器學習有更深一層的認識。