实验五

崔士强 PB22151743

实验三数据

下面使用实验三的数据，对 `diagnosis` 进行预测。

数据读取及预处理

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          568 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   567 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               567 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           568 non-null    float64
 18  concavity_se             568 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     568 non-null    float64
 22  radius_worst             568 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         568 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
None
```

```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mea
n  \
0    842302         M        17.99         10.38          122.80     1001.
0
1    842517         M        20.57         17.77          132.90     1326.
0
2  84300903         M        19.69         21.25          130.00     1203.
0
3  84348301         M        11.42         20.38           77.58      386.
1
4  84358402         M        20.29         14.34          135.10     1297.
0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean
\
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   ...  radius_worst  texture_worst  perimeter_worst  area_worst  \
0  ...         25.38          17.33           184.60      2019.0
1  ...         24.99          23.41           158.80      1956.0
2  ...         23.57          25.53           152.50      1709.0
3  ...         14.91          26.50            98.87       567.7
4  ...         22.54          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave points_wo
rst  \
0            0.1622             0.6656           0.7119                0.2
654
1            0.1238             0.1866           0.2416                0.1
860
2            0.1444             0.4245           0.4504                0.2
430
3            0.2098             0.8663           0.6869                0.2
575
4            0.1374             0.2050           0.4000                0.1
625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678

[5 rows x 32 columns]
```

主实验

对于主实验，采取下面两种算法

1. 支持向量机（SVM）

SVM的目的是找到一个超平面，它可以有效地分离不同类别的数据点，同时尽可能地最大化不同类别之间的间隔。SVM对于小到中等数据集表现良好，尤其是在数据维度较高时。

参考文献：

- Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

1. 梯度提升树（Gradient Boosting Machines, GBM）

GBM通过逐步修正前一个模型的残差来增强模型的预测能力，在处理各种统计分类和回归问题时表现出色。GBM特别适用于处理复杂的非线性关系。

参考文献：

- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of statistics, 1189-1232.

```
SVM Classification Report:
              precision    recall  f1-score   support

           B       0.94      0.98      0.96        63
           M       0.98      0.92      0.95        49

    accuracy                           0.96       112
   macro avg       0.96      0.95      0.95       112
weighted avg       0.96      0.96      0.96       112

GBM Classification Report:
              precision    recall  f1-score   support

           B       0.94      0.95      0.94        63
           M       0.94      0.92      0.93        49

    accuracy                           0.94       112
   macro avg       0.94      0.94      0.94       112
weighted avg       0.94      0.94      0.94       112
```

参数优化

```
Best SVM Parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'linear'}
Best SVM Score: 0.9620224719101124
Best GBM Parameters: {'learning_rate': 0.5, 'max_depth': 3, 'n_estimator
s': 100}
Best GBM Score: 0.966541822721598
SVM Test Set Performance:
              precision    recall  f1-score   support

           B       0.94      1.00      0.97        63
           M       1.00      0.92      0.96        49

    accuracy                           0.96       112
   macro avg       0.97      0.96      0.96       112
weighted avg       0.97      0.96      0.96       112

GBM Test Set Performance:
              precision    recall  f1-score   support

           B       0.94      0.95      0.94        63
           M       0.94      0.92      0.93        49

    accuracy                           0.94       112
   macro avg       0.94      0.94      0.94       112
weighted avg       0.94      0.94      0.94       112
```

与之前的结果基本一致。

实验四数据

下面使用实验四的数据，对`Class`进行预测。

数据读取及预处理

对数据进行预处理，过程包括去除缺失值，转换为独热向量等

```
Columns with missing values:
['node-caps', 'breast-quad']
Value distribution for 'tumor-size' before corrections:
tumor-size
30-34     57
25-29     51
20-24     48
15-19     29
14-Oct    28
40-44     22
35-39     19
0-4        8
50-54      8
9-May      4
45-49      3
Name: count, dtype: int64
```

```
Value distribution for 'inv-nodes' before corrections:
inv-nodes
0-2       209
5-Mar      34
8-Jun      17
11-Sep      7
15-17       6
14-Dec      3
24-26       1
Name: count, dtype: int64

Corrected value distribution for 'tumor-size':
tumor-size
30-34    57
25-29    51
20-24    48
15-19    29
10-14    28
40-44    22
35-39    19
0-4       8
50-54     8
5-9       4
45-49     3
Name: count, dtype: int64

Corrected value distribution for 'inv-nodes':
inv-nodes
0-2      209
3-5       34
6-8       17
9-11       7
15-17      6
12-14      3
24-26      1
Name: count, dtype: int64
0 Class=no-recurrence-events
1 Class=recurrence-events
2 age=10-19
3 age=20-29
4 age=30-39
5 age=40-49
6 age=50-59
7 age=60-69
8 age=70-79
9 age=80-89
10 age=90-99
11 menopause=lt40
12 menopause=ge40
13 menopause=premeno
14 tumor-size=0-4
15 tumor-size=5-9
16 tumor-size=10-14
17 tumor-size=15-19
18 tumor-size=20-24
19 tumor-size=25-29
20 tumor-size=30-34
```

```
21 tumor-size=35-39
22 tumor-size=40-44
23 tumor-size=45-49
24 tumor-size=50-54
25 tumor-size=55-59
26 inv-nodes=0-2
27 inv-nodes=3-5
28 inv-nodes=6-8
29 inv-nodes=9-11
30 inv-nodes=12-14
31 inv-nodes=15-17
32 inv-nodes=18-20
33 inv-nodes=21-23
34 inv-nodes=24-26
35 inv-nodes=27-29
36 inv-nodes=30-32
37 inv-nodes=33-35
38 inv-nodes=36-39
39 node-caps=yes
40 node-caps=no
41 deg-malig=1
42 deg-malig=2
43 deg-malig=3
44 breast=left
45 breast=right
46 breast-quad=left_up
47 breast-quad=left_low
48 breast-quad=right_up
49 breast-quad=right_low
50 breast-quad=central
51 irradiat=yes
52 irradiat=no
      id  Class  age  menopause  tumor-size  inv-nodes  node-caps  deg-mal
ig  \
0      0      0    4         13          20         26         40
43
1      1      0    5         13          18         26         40
42
2      2      0    5         13          18         26         40
42
3      3      0    7         12          17         26         40
42
4      4      0    5         13          14         26         40
42
..   ...    ...  ...        ...         ...        ...        ...
...
281  281      1    4         13          20         26         40
42
282  282      1    4         13          18         26         40
43
283  283      1    7         12          18         26         40
41
284  284      1    5         12          20         27         40
43
285  285      1    6         12          20         27         40
43
```

```
     breast  breast-quad  irradiat
0        44           47        52
1        45           48        52
2        44           47        52
3        45           46        52
4        45           49        52
..      ...          ...       ...
281      44           46        52
282      44           46        51
283      45           46        52
284      44           47        52
285      44           47        52

[277 rows x 11 columns]
```

主实验

我们选择两种算法模型进行比较：

- 逻辑回归：广泛用于二分类问题的线性模型，它预测的是观察属于特定类别的概率。参考资料: Hosmer, D. W., & Lemeshow, S. (2000). Applied Logistic Regression.
- 随机森林：一个基于决策树的集成学习算法，通过构建多棵树并取它们的多数投票来提高预测准确性和稳定性。参考资料: Breiman, L. (2001). Random Forests. Machine Learning.

我们使用所有可用特征，因为逻辑回归和随机森林能够较好地处理高维数据，并从中选择重要的特征。

在这一步，第一次尝试的结果是准确度达到100%（在这里发现的原因是我先做了实验四数据），更换多种方法及模型均无果。最终发现原因是没有把id从特征中去除。

```
逻辑回归模型评估结果:
              precision    recall  f1-score   support

           0       0.73      0.95      0.82        37
           1       0.75      0.32      0.44        19

    accuracy                           0.73        56
   macro avg       0.74      0.63      0.63        56
weighted avg       0.74      0.73      0.69        56

随机森林模型评估结果:
              precision    recall  f1-score   support

           0       0.69      0.78      0.73        37
           1       0.43      0.32      0.36        19

    accuracy                           0.62        56
   macro avg       0.56      0.55      0.55        56
weighted avg       0.60      0.62      0.61        56
```

尝试使用k折交叉验证，结果如下

```
逻辑回归交叉验证评估结果:
fit_time: 0.006 (+/- 0.009)
score_time: 0.002 (+/- 0.000)
test_accuracy: 0.736 (+/- 0.039)
test_precision: 0.720 (+/- 0.044)
test_recall: 0.736 (+/- 0.039)
test_f1: 0.702 (+/- 0.039)

随机森林交叉验证评估结果:
fit_time: 0.041 (+/- 0.001)
score_time: 0.005 (+/- 0.001)
test_accuracy: 0.736 (+/- 0.027)
test_precision: 0.717 (+/- 0.040)
test_recall: 0.736 (+/- 0.027)
test_f1: 0.715 (+/- 0.049)
```

参数优化

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
最佳参数组合: {'max_depth': 10, 'n_estimators': 50}
最佳交叉验证分数: 0.7919289152165865
最佳模型评估结果:
              precision    recall  f1-score   support

           0       0.69      0.84      0.76        37
           1       0.45      0.26      0.33        19

    accuracy                           0.64        56
   macro avg       0.57      0.55      0.54        56
weighted avg       0.61      0.64      0.61        56
```

略有提升。