homework_cancer

February 5, 2024

1 Mô tả bài toán và dữ liệu

Để dự đoán một bệnh nhân có khả năng ung thư hay không, người ta thu thập các mẫu sinh thiết, xét nghiệm bằng việc lấy một lát cắt của mẫu để soi dưới kính hiển vi, quét và ghi lại mẫu dưới dạng các khung ảnh số của các nhân tế bào.

Từ đó, người ta tiến hành đo kích thước, hình dạng, kết cấu của chúng và tính toán 10 đặc tính của nhân. Cụ thể trong tệp dữ liệu các trường sẽ là:
1. Mã số mẫu
2. Loại: 2 cho lành tính, 4 cho ác tính
3. Độ dày hạch (Clump Thickness)
4. Sự đồng nhất của kích thước tế bào (Uniformity of Cell Size)
5. Tính đồng nhất của hình dạng tế bào (Uniformity of Cell Shape)
6. Độ kết dính lề - biên (Marginal Adhesion)
7. Kích thước tế bào biểu mô đơn (Single Epithelial Cell Size)
8. Phần nhân ngoài (Bare Nuclei)
9. Chất nhiễm sắc thể (Bland Chromatin)
10. Nhân trong thường (Normal Nucleoli)
11. Vỏ (Mitoses)

2 Xử lý dữ liệu

Chọn 80 mẫu lành tính (Trường phân loại là 2) và 40 mẫu ác tính (Trường phân loại là 4) làm dữ liệu Test, chúng ta sẽ không sử dụng trường phân loại.

Phần còn lại sẽ là dữ liệu training.

[ ]:
```python
# các hàm viết để phục vụ việc xử lý dữ liệu
import random
import pandas as pd

def read_data(data:str='data/cancer/datacum'):
    '''
    Function to read data from source file data
    '''
    with open(data) as f:
        content = f.readlines()
```

```
    X_data, y_label = [], []

    for line in content:
        line = line.strip()

        if line == '' or line.startswith('#'):
            continue

        parts = line.split(',')
        label = False if parts[1] == '2' else True    # 2 is benign, 4 is␣
 ↪malignant -> 2 is False, 4 is True
        data_input = []
        for index in range(11):
            if index == 1:
                continue
            data_input.append(int(parts[index]))
        X_data.append(data_input)
        y_label.append(label)

    return X_data, y_label

def __get_random_samples__(X_data:list, y_label:list, target_label:bool,␣
 ↪num_samples:int):
    '''
    Function to get random samples
    '''
    indices = [i for i, label in enumerate(y_label) if label == target_label]
    selected_indices = random.sample(indices, num_samples)

    selected_X_data = [X_data[i] for i in selected_indices]
    selected_y_label = [y_label[i] for i in selected_indices]

    return selected_X_data, selected_y_label, selected_indices

def split_train_test_dataset(X_data:list, y_label:list):
    '''
    Function to split dataset to train set and test set
    '''
    column_names = ['Sample code number', 'Clump Thickness', 'Uniformity of␣
 ↪Cell Size', 'Uniformity of Cell Shape', 'Marginal Adhesion', 'Single␣
 ↪Epithelial Cell Size', 'Bare Nuclei', 'Bland Chromatin', 'Normal Nucleoli',␣
 ↪'Mitoses']

    ## Make test set
    # Lấy ngẫu nhiên 80 mẫu có label 2
    X_data_label_2, y_label_2, indices_2 = __get_random_samples__(X_data,␣
 ↪y_label, target_label=False, num_samples=80)
```

```
    # Lấy ngẫu nhiên 40 mẫu có label 4
    X_data_label_4, y_label_4, indices_4 = __get_random_samples__(X_data,␣
↪y_label, target_label=True, num_samples=40)

    # Kết hợp dữ liệu
    test_X_data = X_data_label_2 + X_data_label_4
    test_y_label = y_label_2 + y_label_4
    test_X_data = pd.DataFrame(test_X_data, columns=column_names)
    test_y_label = pd.DataFrame(test_y_label, columns=['Class'])
    test_data = (test_X_data, test_y_label)


  ␣
↪#############################################################################################

    ## Make train set
    train_set_indices = set(range(len(X_data))) - set(indices_2) -␣
↪set(indices_4)

    # Lấy mẫu từ phần còn lại của bộ dữ liệu
    train_X_data = [X_data[i] for i in train_set_indices]
    train_y_label = [y_label[i] for i in train_set_indices]
    train_X_data = pd.DataFrame(train_X_data, columns=column_names)
    train_y_label = pd.DataFrame(train_y_label, columns=['Class'])
    train_data = (train_X_data, train_y_label)

    return train_data, test_data
```

[ ]:
```
# read data
filename = 'data/cancer/datacum'

X_data, y_label = read_data(filename)
```

[ ]:
```
# some view on data
print(len(X_data), len(y_label))
print(X_data[:5])
print(y_label[:5])
```

```
699 699
[[1000025, 5, 1, 1, 1, 2, 1, 3, 1, 1], [1002945, 5, 4, 4, 5, 7, 10, 3, 2, 1],
[1015425, 3, 1, 1, 1, 2, 2, 3, 1, 1], [1016277, 6, 8, 8, 1, 3, 4, 3, 7, 1],
[1017023, 4, 1, 1, 3, 2, 1, 3, 1, 1]]
[False, False, False, False, False]
```

[ ]:
```
# take train and test set
train_data, test_data = split_train_test_dataset(X_data, y_label)
```

[ ]:
```
# some view on train and test dataset
print(len(train_data[0]), len(train_data[1]), train_data)
print(len(test_data[0]), len(test_data[1]), test_data)
```

```
579 579 (     Sample code number  Clump Thickness  Uniformity of Cell Size  \
0                1000025                5                        1
1                1002945                5                        4
2                1015425                3                        1
3                1016277                6                        8
4                1017023                4                        1
..                   …              …                        …
574               763235                3                        1
575               776715                3                        1
576               888820                5                       10
577               897471                4                        8
578               897471                4                        8

     Uniformity of Cell Shape  Marginal Adhesion  Single Epithelial Cell Size  \
0                           1                  1                            2
1                           4                  5                            7
2                           1                  1                            2
3                           8                  1                            3
4                           1                  3                            2
..                        …                  …                            …
574                         1                  1                            2
575                         1                  1                            3
576                        10                  3                            7
577                         6                  4                            3
578                         8                  5                            4

     Bare Nuclei  Bland Chromatin  Normal Nucleoli  Mitoses
0              1                3                1        1
1             10                3                2        1
2              2                3                1        1
3              4                3                7        1
4              1                3                1        1
..           …                …                …      …
574            1                2                1        2
575            2                1                1        1
576            3                8               10        2
577            4               10                6        1
578            5               10                4        1

[579 rows x 10 columns],      Class
0    False
1    False
2    False
3    False
```

```
4    False
..     …
574  False
575  False
576   True
577   True
578   True

[579 rows x 1 columns])
120 120 (     Sample code number  Clump Thickness  Uniformity of Cell Size  \
0               1280258                4                        1
1               1288608                3                        1
2               1272166                5                        1
3               1033078                2                        1
4               1238915                5                        1
..                  …                …                        …
115             1258549                9                       10
116             1076352                3                        6
117             1227210               10                        5
118             1294562               10                        8
119             1176881                7                        5

     Uniformity of Cell Shape  Marginal Adhesion  Single Epithelial Cell Size  \
0                           1                  1                            2
1                           1                  1                            2
2                           1                  1                            2
3                           1                  1                            2
4                           2                  1                            2
..                        …                  …                            …
115                        10                 10                           10
116                         4                 10                            3
117                         5                  6                            3
118                        10                  1                            3
119                         3                  7                            4

     Bare Nuclei  Bland Chromatin  Normal Nucleoli  Mitoses
0              1                1                2        1
1              1                2                1        1
2              1                1                1        1
3              1                1                1        5
4              1                3                1        1
..           …                …                …      …
115           10               10               10        1
116            3                3                4        1
117           10                7                9        2
118           10                5                1        1
119           10                7                5        5
```

```
[120 rows x 10 columns],      Class
0    False
1    False
2    False
3    False
4    False
..     …
115   True
116   True
117   True
118   True
119   True

[120 rows x 1 columns])
```

3 Xây dựng mô hình phân loại Naive Bayes cho dữ liệu liên tục

[ ]:
```
from sklearn.naive_bayes import GaussianNB

# Tạo mô hình Naive Bayes
model = GaussianNB()

# Huấn luyện mô hình trên tập training
model.fit(train_data[0], train_data[1])
```

```
/home/harito/venv/py/lib/python3.11/site-
packages/sklearn/utils/validation.py:1111: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[ ]:
```
GaussianNB()
```

4 Thực hiện kiểm chứng qua các chỉ số Accuracy, Precision, Recall

[ ]:
```
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report

# Dự đoán trên tập test
y_pred = model.predict(test_data[0])

# Đánh giá mô hình sử dụng thư viện
print('Đánh giá mô hình sử dụng thư viện')
accuracy = accuracy_score(test_data[1], y_pred)
report = classification_report(test_data[1], y_pred)
```

```python
print(f"Accuracy: {accuracy}")
print("Classification Report:")
print(report)
# Accuracy: 0.925
#               precision    recall
#       False       0.92      0.97
#        True       0.94      0.82

# Đánh giá mô hình thông qua tính thủ công
print('Đánh giá mô hình thông qua tính thủ công')
conf_matrix = confusion_matrix(test_data[1], y_pred)
print('Confusion matrix:')
print(conf_matrix)
print('Accuracy: ', (conf_matrix[0][0] + conf_matrix[1][1]) / 120)
print('Recall dự đoán False:', (conf_matrix[0][0]) / (conf_matrix[0][0] +␣
 ↪conf_matrix[0][1]))
print('Recall dự đoán True:', (conf_matrix[1][1]) / (conf_matrix[1][0] +␣
 ↪conf_matrix[1][1]))
print('Precision dự đoán False:', (conf_matrix[0][0]) / (conf_matrix[0][0] +␣
 ↪conf_matrix[1][0]))
print('Precision dự đoán True:', (conf_matrix[1][1]) / (conf_matrix[0][1] +␣
 ↪conf_matrix[1][1]))
```

```
Đánh giá mô hình sử dụng thư viện
Accuracy: 0.925
Classification Report:
              precision    recall  f1-score   support

       False       0.92      0.97      0.95        80
        True       0.94      0.82      0.88        40

    accuracy                           0.93       120
   macro avg       0.93      0.90      0.91       120
weighted avg       0.93      0.93      0.92       120

Đánh giá mô hình thông qua tính thủ công
Confusion matrix:
[[78  2]
 [ 7 33]]
Accuracy:  0.925
Recall dự đoán False: 0.975
Recall dự đoán True: 0.825
Precision dự đoán False: 0.9176470588235294
Precision dự đoán True: 0.9428571428571428
```