Muhammad_Dastgir_Breast_Cancer

November 11, 2024

[2]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from ucimlrepo import fetch_ucirepo
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LogisticRegressionCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import random
```

0.1 Loading Dataset from UC Irvine Machine Learning Repository and Assigning Values

[5]:

```
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets
```

[6]:

```
X
```

[6]:

```
     radius1  texture1  perimeter1   area1  smoothness1  compactness1  \
0      17.99     10.38      122.80  1001.0      0.11840       0.27760
1      20.57     17.77      132.90  1326.0      0.08474       0.07864
2      19.69     21.25      130.00  1203.0      0.10960       0.15990
3      11.42     20.38       77.58   386.1      0.14250       0.28390
4      20.29     14.34      135.10  1297.0      0.10030       0.13280
..       …         …           …      …           …             …
564    21.56     22.39      142.00  1479.0      0.11100       0.11590
565    20.13     28.25      131.20  1261.0      0.09780       0.10340
566    16.60     28.08      108.30   858.1      0.08455       0.10230
567    20.60     29.33      140.10  1265.0      0.11780       0.27700
568     7.76     24.54       47.92   181.0      0.05263       0.04362

     concavity1  concave_points1  symmetry1  fractal_dimension1  …  radius3  \
0       0.30010          0.14710     0.2419             0.07871  …   25.380
1       0.08690          0.07017     0.1812             0.05667  …   24.990
2       0.19740          0.12790     0.2069             0.05999  …   23.570
3       0.24140          0.10520     0.2597             0.09744  …   14.910
4       0.19800          0.10430     0.1809             0.05883  …   22.540
```

```
..          …              …        …                  …  …      …
564     0.24390          0.13890     0.1726             0.05623  …   25.450
565     0.14400          0.09791     0.1752             0.05533  …   23.690
566     0.09251          0.05302     0.1590             0.05648  …   18.980
567     0.35140          0.15200     0.2397             0.07016  …   25.740
568     0.00000          0.00000     0.1587             0.05884  …    9.456

     texture3  perimeter3   area3  smoothness3  compactness3  concavity3   \
0       17.33      184.60  2019.0      0.16220       0.66560      0.7119
1       23.41      158.80  1956.0      0.12380       0.18660      0.2416
2       25.53      152.50  1709.0      0.14440       0.42450      0.4504
3       26.50       98.87   567.7      0.20980       0.86630      0.6869
4       16.67      152.20  1575.0      0.13740       0.20500      0.4000
..        …         …     …          …           …         …
564     26.40      166.10  2027.0      0.14100       0.21130      0.4107
565     38.25      155.00  1731.0      0.11660       0.19220      0.3215
566     34.12      126.70  1124.0      0.11390       0.30940      0.3403
567     39.42      184.60  1821.0      0.16500       0.86810      0.9387
568     30.37       59.16   268.6      0.08996       0.06444      0.0000

     concave_points3  symmetry3  fractal_dimension3
0             0.2654     0.4601             0.11890
1             0.1860     0.2750             0.08902
2             0.2430     0.3613             0.08758
3             0.2575     0.6638             0.17300
4             0.1625     0.2364             0.07678
..               …        …                 …
564           0.2216     0.2060             0.07115
565           0.1628     0.2572             0.06637
566           0.1418     0.2218             0.07820
567           0.2650     0.4087             0.12400
568           0.0000     0.2871             0.07039

[569 rows x 30 columns]
```

[7]:

```
y
```

[7]:

```
    Diagnosis
0           M
1           M
2           M
3           M
4           M
..        …
564         M
565         M
566         M
```

```
567          M
568          B

[569 rows x 1 columns]
```

0.1.1 Creating training and testing data

[8]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,␣
  ↪random_state = 69)
```

[9]:
```
training_model = LogisticRegression()
training_model.fit(X_train, y_train)
```

```
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/utils/validation.py:1184: 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)
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[9]:
```
LogisticRegression()
```

[10]:
```
predicted_y = training_model.predict(X_test)
predicted_y # for reference
```

[10]:
```
array(['M', 'B', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B',
       'M', 'B', 'B', 'M', 'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'M',
       'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
       'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'M', 'B', 'B', 'M',
       'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
       'B', 'M', 'B', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B',
       'M', 'M', 'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'M', 'B', 'M',
       'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'M',
       'M', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'M'], dtype=object)
```

[11]:
```
accuracy = training_model.score(X_test, y_test)
accuracy
```

[11]:
```
0.9298245614035088
```

[12]:
```
Cs = np.logspace(-4, 4, 10)
```

[13]:
```
l1_model = LogisticRegressionCV(
    Cs=Cs,
    cv=5,
    penalty='l1',
    solver='liblinear',
    max_iter=1000,
    scoring='accuracy',
    refit=True
)
```

[14]:
```
l1_model.fit(X_train, y_train)
```

```
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/utils/validation.py:1184: 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)
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/svm/_base.py:1242: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/svm/_base.py:1242: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/svm/_base.py:1242: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/svm/_base.py:1242: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/svm/_base.py:1242: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/svm/_base.py:1242: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
```

[14]:
```
LogisticRegressionCV(Cs=array([1.00000000e-04, 7.74263683e-04, 5.99484250e-03,
4.64158883e-02,
       3.59381366e-01, 2.78255940e+00, 2.15443469e+01, 1.66810054e+02,
       1.29154967e+03, 1.00000000e+04]),
                     cv=5, max_iter=1000, penalty='l1', scoring='accuracy',
                     solver='liblinear')
```

[15]:

```
l2_model = LogisticRegressionCV(
    Cs=Cs,
    cv=5,
    penalty='l2',
    solver='lbfgs',
    max_iter=1000,
    scoring='accuracy',
    refit=True
)
```

[16]:

```
l2_model.fit(X_train, y_train)
```

```
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/utils/validation.py:1184: 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)
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[16]:

```
LogisticRegressionCV(Cs=array([1.00000000e-04, 7.74263683e-04, 5.99484250e-03,
4.64158883e-02,
       3.59381366e-01, 2.78255940e+00, 2.15443469e+01, 1.66810054e+02,
       1.29154967e+03, 1.00000000e+04]),
                     cv=5, max_iter=1000, scoring='accuracy')
```

[17]:

```
l2_model.fit(X_train, y_train)
```

```
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/utils/validation.py:1184: 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)
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/hamiddastgir/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
```

```
regression
  n_iter_i = _check_optimize_result(
```

[17]:

```
LogisticRegressionCV(Cs=array([1.00000000e-04, 7.74263683e-04, 5.99484250e-03,
4.64158883e-02,
       3.59381366e-01, 2.78255940e+00, 2.15443469e+01, 1.66810054e+02,
       1.29154967e+03, 1.00000000e+04]),
                     cv=5, max_iter=1000, scoring='accuracy')
```

[18]:

```python
optimal_C_l1 = l1_model.C_[0]
optimal_C_l2 = l2_model.C_[0]

print(f'Optimal C for L1 penalty: {optimal_C_l1}')
print(f'Optimal C for L2 penalty: {optimal_C_l2}')
```

```
Optimal C for L1 penalty: 21.54434690031882
Optimal C for L2 penalty: 1291.5496650148827
```

[19]:

```python
y_pred_l1 = l1_model.predict(X_test)
y_pred_l2 = l2_model.predict(X_test)

accuracy_l1 = accuracy_score(y_test, y_pred_l1)
accuracy_l2 = accuracy_score(y_test, y_pred_l2)

print(f'L1 Penalty Accuracy: {accuracy_l1:.2f}')
print(f'L2 Penalty Accuracy: {accuracy_l2:.2f}')
```

```
L1 Penalty Accuracy: 0.95
L2 Penalty Accuracy: 0.93
```

[20]:

```python
n_features_l1 = np.sum(l1_model.coef_ != 0)
n_features_l2 = np.sum(l2_model.coef_ != 0)

print(f'Number of features used in L1 model: {n_features_l1}')
print(f'Number of features used in L2 model: {n_features_l2}')
```

```
Number of features used in L1 model: 18
Number of features used in L2 model: 30
```

0.2 Analysis

The lasso regularized model was more accurate, and it had more non-zero coefficients. The lasso regularized model also benefitted from a higher lamda (inverse C - used above)

[21]:

```python
y_pred_l1
```

[21]:

```
array(['M', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B',
       'M', 'B', 'M', 'B', 'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'M',
       'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
       'M', 'B', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'B', 'M',
       'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
       'B', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'B',
       'M', 'M', 'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'M', 'B', 'M',
       'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'M', 'M', 'M', 'B', 'M',
       'M', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'M'], dtype=object)
```

```
[22]: y_pred_l2
```

```
[22]: array(['M', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B',
       'M', 'B', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'M',
       'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
       'M', 'B', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'B', 'M',
       'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M',
       'B', 'M', 'B', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'B', 'M', 'B',
       'M', 'M', 'B', 'M', 'M', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'M',
       'B', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'M',
       'M', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'M'], dtype=object)
```