my logistic regression for ayebale peter bcs 00109

March 19, 2024

[6]:
```
#library importation
import numpy as np
import pandas as pd
```

[7]:
```
myset=pd.read_csv("C:/Users/hj/Desktop/trainingz.csv")
myset
```

[7]:
```
      No  Survived  Pclass     Sex   Age  SibSp      Fare
0      1         0       1    male  50.0      1   55.9000
1      2         0       3    male  40.5      0    7.7500
2      3         0       1    male  45.0      0   26.5500
3      4         0       2    male  32.0      0   10.5000
4      5         0       1    male  47.0      0   52.0000
..   …        …      …      …   …    …       …
707  708         1       2  female  24.0      0   14.5000
708  709         1       3    male  30.0      0    9.5000
709  710         1       1  female  43.0      0  211.3375
710  711         1       1    male  42.0      0   26.2875
711  712         1       2  female  27.0      0   10.5000

[712 rows x 7 columns]
```

[8]:
```
#droping the target variables
x=myset.drop(columns=["Sex"])
x
```

[8]:
```
      No  Survived  Pclass   Age  SibSp      Fare
0      1         0       1  50.0      1   55.9000
1      2         0       3  40.5      0    7.7500
2      3         0       1  45.0      0   26.5500
3      4         0       2  32.0      0   10.5000
4      5         0       1  47.0      0   52.0000
..   …        …      …    …   …      …
707  708         1       2  24.0      0   14.5000
708  709         1       3  30.0      0    9.5000
709  710         1       1  43.0      0  211.3375
710  711         1       1  42.0      0   26.2875
711  712         1       2  27.0      0   10.5000
```

```
[712 rows x 6 columns]
```

[9]:

```
#extraction of a target variable
y=myset["Sex"].values
y
```

[9]:

```
array(['male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'female',
       'female', 'male', 'male', 'male', 'female', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'female', 'male', 'female', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'female', 'male', 'female', 'male',
       'male', 'female', 'male', 'male', 'male', 'male', 'female',
       'female', 'female', 'male', 'female', 'male', 'male', 'female',
       'male', 'female', 'male', 'female', 'male', 'female', 'male',
       'female', 'female', 'male', 'male', 'female', 'male', 'female',
       'male', 'female', 'female', 'female', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'female', 'male', 'male', 'male', 'male', 'male', 'male', 'female',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'female',
       'female', 'male', 'male', 'female', 'male', 'male', 'female',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'female', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'female', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'female', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'female', 'female', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'female',
       'female', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'female', 'female', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'female', 'female', 'male', 'male',
       'male', 'male', 'male', 'female', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'female', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'female', 'male', 'male', 'male', 'female', 'male', 'male',
       'male', 'male', 'female', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'female', 'male', 'male', 'male',
       'male', 'male', 'male', 'female', 'male', 'male', 'male', 'male',
       'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'female', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'male', 'female', 'male', 'female', 'male', 'male',
'male', 'male', 'female', 'male', 'female', 'male', 'male', 'male',
'male', 'male', 'male', 'female', 'male', 'male', 'male', 'male',
'male', 'male', 'male', 'male', 'male', 'female', 'female', 'male',
'female', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'male', 'male', 'female', 'male', 'male', 'female',
'male', 'male', 'male', 'female', 'female', 'male', 'male', 'male',
'male', 'male', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'female', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'female', 'male', 'male', 'female', 'male', 'male',
'male', 'female', 'male', 'male', 'male', 'male', 'male', 'male',
'male', 'male', 'male', 'female', 'male', 'male', 'male', 'male',
'female', 'male', 'male', 'male', 'male', 'female', 'female',
'male', 'female', 'female', 'male', 'female', 'female', 'female',
'female', 'female', 'female', 'female', 'male', 'female', 'female',
'male', 'male', 'male', 'female', 'male', 'female', 'female',
'female', 'female', 'female', 'male', 'male', 'female', 'male',
'female', 'female', 'female', 'female', 'male', 'male', 'female',
'female', 'female', 'female', 'male', 'female', 'female', 'male',
'female', 'female', 'female', 'male', 'female', 'male', 'female',
'female', 'female', 'female', 'female', 'female', 'female',
'female', 'female', 'female', 'female', 'female', 'female', 'male',
'female', 'female', 'male', 'female', 'male', 'female', 'male',
'female', 'male', 'female', 'female', 'female', 'female', 'female',
'male', 'female', 'female', 'female', 'female', 'female', 'female',
'male', 'female', 'female', 'male', 'male', 'male', 'female',
'male', 'male', 'female', 'female', 'male', 'male', 'male',
'female', 'male', 'female', 'male', 'male', 'male', 'female',
'male', 'female', 'female', 'male', 'male', 'female', 'male',
'female', 'female', 'female', 'female', 'male', 'female', 'female',
'female', 'female', 'female', 'female', 'male', 'female', 'female',
'male', 'female', 'female', 'male', 'female', 'female', 'female',
'female', 'female', 'female', 'male', 'female', 'female', 'female',
'female', 'female', 'female', 'male', 'female', 'female', 'female',
'female', 'female', 'female', 'female', 'female', 'male', 'female',
'male', 'female', 'male', 'female', 'female', 'male', 'male',
'male', 'male', 'female', 'male', 'male', 'female', 'male',
'female', 'male', 'male', 'female', 'male', 'female', 'male',
'female', 'female', 'female', 'female', 'female', 'female',
'female', 'female', 'female', 'male', 'female', 'female', 'female',
'female', 'male', 'male', 'female', 'female', 'male', 'female',
'female', 'female', 'female', 'female', 'female', 'male', 'male',
'male', 'male', 'female', 'female', 'male', 'female', 'male',
'female', 'female', 'female', 'female', 'female', 'male', 'female',
'female', 'female', 'male', 'female', 'female', 'female', 'male',
'female', 'female', 'male', 'female', 'female', 'male', 'male',
'female', 'female', 'female', 'female', 'female', 'male', 'female',
'female', 'male', 'female', 'female', 'female', 'female', 'female',
'female', 'male', 'female', 'female', 'female', 'male', 'female',
'female', 'female', 'female', 'female', 'female', 'female',
'female', 'male', 'male', 'female', 'female', 'male', 'female',
'male', 'female', 'female', 'female', 'male', 'female', 'male',
'female'], dtype=object)
```

```
[10]: #checking missing data
      myset.isna().sum()
```

```
[10]: No          0
      Survived    0
      Pclass      0
      Sex         0
      Age         0
      SibSp       0
      Fare        0
      dtype: int64
```

```
[11]: #let me split my data
      from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
       ↪2,random_state=48)
```

```
[12]: #creating and fitting a model
      from sklearn.linear_model import LogisticRegression
```

```
[13]: model=LogisticRegression()
      model
```

```
[13]: LogisticRegression()
```

```
[14]: model.fit(x_train,y_train)
```

```
[14]: LogisticRegression()
```

```
[15]: #predictions
      y_pred=model.predict(x_test)
      y_pred
```

```
[15]: array(['male', 'male', 'male', 'male', 'female', 'male', 'male', 'female',
             'female', 'male', 'female', 'male', 'male', 'male', 'female',
             'male', 'male', 'male', 'male', 'male', 'female', 'female', 'male',
             'female', 'male', 'male', 'female', 'female', 'female', 'female',
             'male', 'male', 'male', 'male', 'male', 'male', 'male', 'female',
             'female', 'male', 'male', 'male', 'male', 'male', 'female', 'male',
             'male', 'male', 'male', 'female', 'male', 'male', 'female', 'male',
             'male', 'female', 'female', 'male', 'female', 'female', 'male',
             'female', 'male', 'female', 'female', 'female', 'male', 'female',
             'female', 'male', 'male', 'female', 'female', 'male', 'female',
             'male', 'female', 'male', 'male', 'male', 'male', 'male', 'male',
             'female', 'female', 'female', 'female', 'male', 'male', 'female',
             'male', 'male', 'male', 'male', 'male', 'male', 'male', 'female',
             'male', 'male', 'female', 'male', 'male', 'male', 'male', 'male',
             'male', 'male', 'male', 'male', 'male', 'female', 'female',
             'female', 'male', 'male', 'female', 'male', 'male', 'female',
             'male', 'male', 'male', 'female', 'male', 'female', 'male', 'male',
             'male', 'female', 'female', 'female', 'female', 'female', 'female',
             'male', 'female', 'male', 'female', 'male', 'female', 'male',
             'male'], dtype=object)
```

[16]:
```
#getting coefficients,intercept
cof=model.coef_
interc=model.intercept_
```

[17]:
```
cof
```

[17]:
```
array([[ 3.33386237e-04, -2.53788362e+00, -1.26199064e-01,
         3.15439018e-03, -2.65254052e-01, -1.56116866e-03]])
```

[18]:
```
interc
```

[18]:
```
array([1.94067091])
```

[19]:
```
#evaluating the model
from sklearn.metrics import accuracy_score,precision_score
```

[20]:
```
#uniques values in ypred and ytest
unitest=np.unique(y_test)
unipred=np.unique(y_pred)
```

[21]:
```
unitest
```

[21]:
```
array(['female', 'male'], dtype=object)
```

[22]:
```
unipred
```

[22]:
```
array(['female', 'male'], dtype=object)
```

[23]:
```
#computing accuracy score
acc=accuracy_score(y_test,y_pred)
acc
```

[23]:
```
0.8111888111888111
```

[24]:
```
#precision score
prec=precision_score(y_test,y_pred,pos_label='male')
prec
```

[24]:
```
0.9101123595505618
```

[25]:
```
#optimizing the model
from sklearn.model_selection import GridSearchCV
```

[26]:
```
model=LogisticRegression()
model
```

[26]:
```
LogisticRegression()
```

[27]:
```
#parameters
param_grid={
    'fit_intercept':[True,False],
    'solver':['liblinear','saga'],
    'penalty':['l1','l2'],

}
param_grid
```

[27]:
```
{'fit_intercept': [True, False],
 'solver': ['liblinear', 'saga'],
 'penalty': ['l1', 'l2']}
```

[28]:
```
#initialise the grid search
gs=GridSearchCV(model,param_grid,cv=5)
gs
```

[28]:
```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'fit_intercept': [True, False],
                         'penalty': ['l1', 'l2'],
                         'solver': ['liblinear', 'saga']})
```

[35]:
```
#performing grid search to find best hyperparameter
gs.fit(x_train,y_train)
```

```
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
D:\TEACHER\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  warnings.warn(
```

[35]:

```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'fit_intercept': [True, False],
                         'penalty': ['l1', 'l2'],
                         'solver': ['liblinear', 'saga']})
```

[37]:

```
#best parameters
bparams=gs.best_params_
bparams
```

[37]:

```
{'fit_intercept': False, 'penalty': 'l2', 'solver': 'liblinear'}
```

[ ]:

[ ]: