MWESIGWA JATIUS LOGISTICS REGRESSION

March 17, 2024

```
[87]: #libraries
      import numpy as np
      import pandas as pd
```

```
[88]: file=pd.read_csv("C:\\Users\\hj\\Desktop\\jat\\world_population.csv")
      file
```

```
[88]:        # Country (or dependency)  Net Change  Density  in p/sqkm  \
      0      1                   India    11454490                  481
      1      2                   China     -215985                  152
      2      3           United States     1706706                   37
      3      4               Indonesia     2032783                  153
      4      5                Pakistan     4660796                  312
      ..   …                       …          …                  …
      228  230              Montserrat          -4                   44
      229  231        Falkland Islands          11                    0
      230  232                    Niue           1                    7
      231  233                 Tokelau          22                  189
      232  234                Holy See           8                 1295

           Land Area  kmsqd  Migrants (net)  Fert. Rate  Med. Age
      0             2973190         -486136       1.999        28
      1             9388211         -310220       1.190        39
      2             9147420          999700       1.662        38
      3             1811570          -49997       2.134        30
      4              770880         -165988       3.347        21
      ..                  …              …          …       …
      228               100               0       1.556        44
      229             12170               0       1.585        40
      230               260               0       2.390        36
      231                10               0       2.635        27
      232                 0               0       2.233        23

      [233 rows x 8 columns]
```

```
[89]: x=file.drop(['Country (or dependency)'],axis=1)
      x
```

[89]:

```
       #  Net Change  Density  in p/sqkm  Land Area  kmsqd  Migrants (net)  \
0      1    11454490                 481            2973190         -486136
1      2     -215985                 152            9388211         -310220
2      3     1706706                  37            9147420          999700
3      4     2032783                 153            1811570          -49997
4      5     4660796                 312             770880         -165988
..   …         …                 …               …             …
228  230          -4                  44                100               0
229  231          11                   0              12170               0
230  232           1                   7                260               0
231  233          22                 189                 10               0
232  234           8                1295                  0               0

     Fert. Rate  Med. Age
0         1.999        28
1         1.190        39
2         1.662        38
3         2.134        30
4         3.347        21
..          …       …
228       1.556        44
229       1.585        40
230       2.390        36
231       2.635        27
232       2.233        23

[233 rows x 7 columns]
```

[134]:

```
y=file['Med. Age']
y.shape
y
```

[134]:

```
0      28
1      39
2      38
3      30
4      21
       ..
228    44
229    40
230    36
231    27
232    23
Name: Med. Age, Length: 233, dtype: int64
```

[135]:

```
from sklearn.model_selection import train_test_split
```

[136]:
```
#training our data
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
```

[137]:
```
from sklearn.linear_model import LogisticRegression
```

[138]:
```
model=LogisticRegression( max_iter=1000,).fit(x_train,y_train)
model
```

```
D:\TEACHER\Lib\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(
```

[138]:
```
LogisticRegression(max_iter=1000)
```

[139]:
```
y_pred=model.predict(x_test)
y_pred
```

[139]:
```
array([28, 27, 43, 54, 28, 28, 40, 28, 28, 30, 43, 28, 49, 32, 22, 32, 32,
       15, 32, 28, 32, 43, 39, 40, 38, 30, 15, 44, 27, 43, 32, 27, 43, 43,
       28, 54, 27, 46, 39, 28, 40, 39, 32, 21, 21, 28, 41], dtype=int64)
```

[140]:
```
from sklearn.metrics import␣
 ↪mean_absolute_error,mean_squared_error,r2_score,accuracy_score
```

[141]:
```
mae=mean_absolute_error(y_test,y_pred)
mae
```

[141]:
```
6.0212765957446805
```

[142]:
```
mse=mean_squared_error(y_test,y_pred)
mse
```

[142]:
```
60.1063829787234
```

[143]:
```
r2=r2_score(y_test,y_pred)
r2
```

[143]:
```
0.30688960347455685
```

[144]:
```
aqsko=accuracy_score(y_test,y_pred)
aqsko
```

[144]: 0.0425531914893617

[145]:
```
from sklearn.model_selection import GridSearchCV
```

[146]:
```
model=LogisticRegression()
param_Grid={
    'penalty':['l1','l2','elasticnet',None],
    'dual':['True','False'],
}
```

[147]:
```
classid=GridSearchCV(model,param_Grid,cv=4)
classid
```

[147]:
```
GridSearchCV(cv=4, estimator=LogisticRegression(),
             param_grid={'dual': ['True', 'False'],
                         'penalty': ['l1', 'l2', 'elasticnet', None]})
```

[169]:
```
#tuning the model
from sklearn.grid_search import GridSearchCV
params = {"n_neighbors": np.arange(1,3),"metric": ["euclidean", "cityblock"]}
grid = GridSearch(estimator=knn, param_grid=params)
grid.fit(x_train, y_train)
print(grid.best_score_)
print(grid.best_estimator_.n_neighbors)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[169], line 2
      1 #tuning the model
----> 2 from sklearn.grid_search import GridSearchCV
      3 params = {"n_neighbors": np.arange(1,3),"metric": ["euclidean",␣
 ↪"cityblock"]}
      4 grid = GridSearch(estimator=knn, param_grid=params)

ModuleNotFoundError: No module named 'sklearn.grid_search'
```

[ ]:
```
```

[ ]:
```
```