

Kabahemba Joanitah BCS,01667-CW 18,March

1

```
#import necessary libraries
import pandas as pd
import numpy as np
#warnings.filterwarnings('ignore')
#load data
df=pd.read_csv("C:\\Users\\Joan k\\Desktop\\datasets\\cancer.csv")
df
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction
Age Outcome
0 6 148 72 35 0 33.6 0.627 50 1
1 1 85 66 29 0 26.6 0.351 31 0
2 8 183 64 0 0 23.3 0.672 32 1
3 1 89 66 23 94 28.1 0.167 21 0
4 0 137 40 35 168 43.1 2.288 33 1
... ..
763 10 101 76 48 180 32.9 0.171 63 0
764 2 122 70 27 0 36.8 0.340 27 0
765 5 121 72 23 112 26.2 0.245 30 0
766 1 126 60 0 0 30.1 0.349 47 1
767 1 93 70 31 0 30.4 0.315 23 0
768 rows x 9 columns

assigning data to x and y axis=1 drops rows

x=df.drop(['Outcome'],axis=1)
x
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction
Age
0 6 148 72 35 0 33.6 0.627 50
1 1 85 66 29 0 26.6 0.351 31
2 8 183 64 0 0 23.3 0.672 32
3 1 89 66 23 94 28.1 0.167 21
4 0 137 40 35 168 43.1 2.288 33
... ..
```

```

763 10 101 76 48 180 32.9 0.171 63
764 2 122 70 27 0 36.8 0.340 27
765 5 121 72 23 112 26.2 0.245 30
766 1 126 60 0 0 30.1 0.349 47
767 1 93 70 31 0 30.4 0.315 23
768 rows × 8 columns

```

```

y=df['Outcome']
y
0      1
1      0
2      1
3      0
4      1
..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
splitting 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=42)
standardize

```

choosing and building the model

```

from sklearn.linear_model import LogisticRegression

```

```

model=LogisticRegression(max_iter=1000)
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
array([0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      dtype=int64)

```

testing the model

```
#y_pred=model.predict([6,148,72,35,0,33.6,0.627,50][1,85,66,29,0,26.6,0.351])
```

```
check for accuracy
```

```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score,accuracy_score
print(mean_absolute_error(y_test,y_pred))
0.2532467532467532
print(mean_squared_error(y_test,y_pred))
0.2532467532467532
print(r2_score(y_test,y_pred))
-0.10303030303030325
#accuracy score
print(accuracy_score(y_test,y_pred))
0.7467532467532467
model=LogisticRegression()
model
```

```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
param_grid={
    "penalty": [None, 'l2'],
    "C": [1.0, 1.5],
    "solver": ['newton-cg', 'newton-cholesky']
}
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(model,param_grid,cv=5,n_jobs=-1)
grid_search.fit(x_train,y_train)
C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:20: FutureWarning:
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.
```

Below are more details about the failures:

```
-----
14 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 114, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py", line 114, in _validate_params
    estimator._validate_params()
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 114, in _validate_params
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'penalty' parameter of LogisticRegression is not an instance of str, int, float, bool, or None.
```

```

-----
4 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 114, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in fit
    estimator._validate_params()
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 114, in _validate_params
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'penalty' parameter of LogisticRegression is not an acceptable value.
-----

```

```

-----
2 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 114, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in fit
    estimator._validate_params()
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 114, in _validate_params
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'penalty' parameter of LogisticRegression is not an acceptable value.
-----

```

```

warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\Joan k\AppData\Local\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:114: UserWarning: Some fits failed with the following error:
nan nan]
warnings.warn(
GridSearchCV(cv=5, estimator=LogisticRegression(), n_jobs=-1,
              param_grid={'C': [1.0, 1.5], 'penalty': [None, 'l2'],
                          'solver': ['newton-cg', 'newton-cholesky']})
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
best_params = grid_search.best_params_
print("Best Parameters :", best_params)
Best Parameters : {'C': 1.0, 'penalty': None, 'solver': 'newton-cg'}
best_model = LogisticRegression(**best_params)
best_model
LogisticRegression(penalty=None, solver='newton-cg')
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
best_model.fit(x_train, y_train)
LogisticRegression(penalty=None, solver='newton-cg')

```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the output. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.

```

best_model
LogisticRegression(penalty=None, solver='newton-cg')
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the output. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.
y_pred = best_model.predict(x_test)
y_pred
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      dtype=int64)
#assessing model fit,
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test,y_pred)
r2_score = r2_score(y_test,y_pred)
accuracy = accuracy_score(y_test,y_pred)
print(f"mean Absolute Error=" ,{mae})
print(f"mean Squared Error=" ,{mse})
print(f"r2_score=" ,{r2_score})
print(f"Accuracy=" ,{accuracy})
mean Absolute Error= {0.24675324675324675}
mean Squared Error= {0.24675324675324675}
r2_score= {-0.074747474747475}
Accuracy= {0.7532467532467533}

```