In [31]:
```
import warnings
warnings.filterwarnings:"ignore"

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression


data =pd.read_csv("C:\\Users\\hm\\Desktop\\LINEAR PROGRAMMING CENTRALS\\Logisti

data

X = np.array(data[["Feature 1","Feature 2"]])
y =np.array(data["Label"])

X

y

# split in data
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.2,random_stat

# standadizing, the independent variables(X VARAIBLES)
scaler=StandardScaler()
X_train_scaled=scaler.fit_transform(X_train)
X_test_scaled=scaler.transform(X_test)


# Building the model
model=LogisticRegression()

model.fit(X_train_scaled,y_train)

y_pred=model.predict(X_test_scaled)

y_pred

accuracy=accuracy_score(y_test,y_pred)

accuracy

# model optimization
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split,GridSearchCV

data =pd.read_csv("C:\\Users\\hm\\Desktop\\LINEAR PROGRAMMING CENTRALS\\Logisti

X = np.array(data[["Feature 1","Feature 2"]])
y =np.array(data["Label"])
```

In [32]:
```
# soliting the data
# split in data
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.2,random_stat
```

In [33]:
```
# building a logistic model
model = LogisticRegression()
```

In [34]:
```
model.fit(X_train, y_train)
```

Out[34]:
```
LogisticRegression
LogisticRegression()
```

In [35]:
```
# building logisticregression model
model= LogisticRegression()
param_grid={
    "C" : [ 0.001,0.01,0.1,1],
    'penalty':['l1','l2'],
    'solver' :['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'sa
   }
```

In [36]:
```
gridsearch = GridSearchCV(model,param_grid, cv=5)
```

In [39]:
```
import warnings
warnings.filterwarnings:"ignore"
gridsearch.fit(X_train,y_train)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not co
nverge
  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_sag.py:350:
ConvergenceWarning: The max_iter was reached which means the coef_ did not co
nverge
  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validatio
n.py:425: FitFailedWarning:
80 fits failed out of a total of 240.
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 erro
r_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
---
20 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_v
alidation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 115
1, in wrapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logi
stic.py", line 1168, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logi
stic.py", line 56, in _check_solver
    raise ValueError(
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penal
ty.

--------------------------------------------------------------------------------
---
20 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_v
alidation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 115
1, in wrapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logi
stic.py", line 1168, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logi
stic.py", line 56, in _check_solver
    raise ValueError(
ValueError: Solver newton-cg supports only 'l2' or 'none' penalties, got l1 p
```

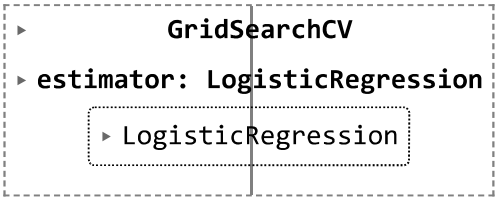
```
enalty.

--------------------------------------------------------------------------------
20 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in wrapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py", line 1168, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py", line 56, in _check_solver
    raise ValueError(
ValueError: Solver newton-cholesky supports only 'l2' or 'none' penalties, got l1 penalty.

--------------------------------------------------------------------------------
20 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in wrapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py", line 1168, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py", line 56, in _check_solver
    raise ValueError(
ValueError: Solver sag supports only 'l2' or 'none' penalties, got l1 penalty.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: UserWarning: One or more of the test scores are non-finite: [   nan 0.425     nan    nan    nan 0.55   0.7625 1.     0.7625 0.7625
 0.7625 0.7625    nan 0.425     nan    nan    nan 0.575  1.     1.
 1.     1.     1.     1.        nan 1.        nan    nan    nan 1.
 1.     1.     1.     1.     1.     1.        nan 1.        nan    nan
    nan 1.     1.     1.     1.     1.     1.     1.    ]
  warnings.warn(
```

Out[39]:



In [38]:
```
best_params = gridsearch.best_params_
```

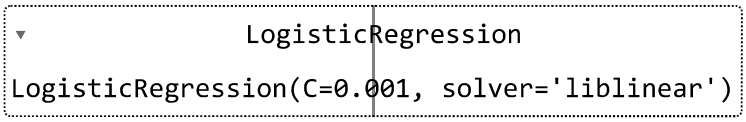
In [20]:
```
print("best parameters: ",best_params)
```

```
best parameters:  {'C': 0.001, 'penalty': 'l2', 'solver': 'liblinear'}
```

In [23]:
```
# getting the best model to train the model
best_model = LogisticRegression(**best_params)
```

In [25]:
```
best_model.fit(X_train, y_train)
```

Out[25]:



In [27]:
```
y_pred = best_model.predict(X_test)
```

In [28]:
```
accuracy = accuracy_score(y_test, y_pred)
print("accuracy: ", accuracy)
```

```
accuracy:  1.0
```

In [ ]: