Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Generate Dataset

```
 make_classification
amples=1000,n_features=5,n_classes=2,n_clusters_per_class=1,random_state=2500)
```

```
X[0:5]
```

```
array([[-1.05725592, -1.30225279, -1.46889882, -1.1266874 ,  1.24135436],
       [ 2.02829207, -1.92846587, -0.9868359 , -1.81084964,  2.0087454 ],
       [ 0.3263314 , -1.01864946, -0.92874165, -0.90770617,  1.00260713],
       [-0.33353226,  0.01998436, -0.69739414,  0.10353936, -0.12231527],
       [ 1.08320214,  2.11279242,  3.03617595,  1.74972077, -1.92032348]])
```

```
y[0:5]
```

```
array([1, 1, 1, 1, 0])
```

```
X.shape, y.shape
```

```
((1000, 5), (1000,))
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_stat
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((700, 5), (300, 5), (700,), (300,))
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(max_iter=500)
```

```
3 lr.fit(X_train,y_train)
```

```
    LogisticRegression(max_iter=500)
```

Model Prediction

```
1 y_pred=lr.predict(X_test)
```

```
1 y_pred.shape
```

```
    (300,)
```

```
1 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
1 accuracy_score(y_test,y_pred)
```

```
    0.9733333333333334
```

```
1 confusion_matrix(y_test,y_pred)
```

```
    array([[152,   4],
           [  4, 140]])
```

```
1 print(classification_report(y_test,y_pred))
```

```
                  precision    recall  f1-score   support

               0       0.97      0.97      0.97       156
               1       0.97      0.97      0.97       144

        accuracy                           0.97       300
       macro avg       0.97      0.97      0.97       300
    weighted avg       0.97      0.97      0.97       300
```

Hyperparameter Tuning

```
1   from  sklearn.model_selection import GridSearchCV
2   parameter={'penalty':['l1','l2'],'C':[0.001,0.009,.09,1,5,10,25],'solver':['libli
3   gridsearch=GridSearchCV(LogisticRegression(),parameter)
4   gridsearch.fit(X_train,y_train)
```

```
    GridSearchCV(estimator=LogisticRegression(),
                 param_grid={'C': [0.001, 0.009, 0.09, 1, 5, 10, 25],
                             'penalty': ['l1', 'l2'], 'solver': ['liblinear']})
```

```
1   gridsearch.best_params
```

```
gridsearch.best_params_
```

```
{'C': 25, 'penalty': 'l1', 'solver': 'liblinear'}
```

```
gridsearch.best_score_
```

```
0.9871428571428572
```

```
gridsearch.best_estimator_
```

```
LogisticRegression(C=25, penalty='l1', solver='liblinear')
```

```
gridsearch.best_index_
```

```
12
```

```
y_pred_grid=gridsearch.predict(X_test)
```

```
confusion_matrix(y_test,y_pred_grid)
```

```
array([[153,   3],
       [  4, 140]])
```

```
print(classification_report(y_test,y_pred_grid))
```

```
                precision    recall  f1-score   support

             0       0.97      0.98      0.98       156
             1       0.98      0.97      0.98       144

      accuracy                           0.98       300
     macro avg       0.98      0.98      0.98       300
  weighted avg       0.98      0.98      0.98       300
```

```
```


0s
completed at 08:27