Assignment - 3 : Logistic Regression

Problem Statement:

Download the iris dataset The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. A.Apply Data pre-processing (Label Encoding, Data Transformation....) techniques if necessary. B.Perform data-preparation (Train-Test Split) C.Apply Logistic Regression Algorithm D.Evaluate Model

importing python libraries

In [1]:

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import *
```

Loading the dataset from seaborn library

In [2]:

```
A=sns.load_dataset("iris")
A
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

Applying label encoding to "species" column

In [3]:

```
from sklearn import preprocessing

# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
A['species']= label_encoder.fit_transform(A['species'])

A['species'].unique()
```

Out[3]:

```
array([0, 1, 2])
```

Assigning independent and dependent variable

In [4]:

```
X=A.iloc[:,[0,1,2,3]].values
Y=A.iloc[:,4].values
```

Dividing the data into training and testing data

In [5]:

```
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=0)
```

feature tranformation

In [6]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()

x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

creating object of Logistic Regression using logistic regression class and fitting the model

In [7]:

```
from sklearn.linear_model import LogisticRegression
```

In [8]:

```
classifier=LogisticRegression()

classifier.fit(x_train,y_train)
```

Out[8]:

```
▾ LogisticRegression
LogisticRegression()
```

predicting the labels of data values

In [9]:

```
x_pred=classifier.predict(x_test)
```

In [10]:

```
x_pred
```

Out[10]:

```
array([2, 1, 0, 2, 0, 2, 0, 2, 2, 1, 2, 2, 1, 2, 2, 0, 2, 1, 0, 0, 2, 2,
       0, 0, 2, 0, 0, 1, 1, 0])
```

Creating confusion matrix

In [11]:

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,x_pred)
print(cm)
```

```
[[11  0  0]
 [ 0  6  7]
 [ 0  0  6]]
```

Calculating accuracy metrics

In [12]:

```
from sklearn.metrics import classification_report

print(classification_report(y_test,x_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      0.46      0.63        13
           2       0.46      1.00      0.63         6

    accuracy                           0.77        30
   macro avg       0.82      0.82      0.75        30
weighted avg       0.89      0.77      0.77        30
```

Calculating accuracy for given model

In [13]:

```
print("Accuracy: ",metrics.accuracy_score(y_test, x_pred))
```

```
Accuracy:  0.7666666666666667
```