**LINEAR DISCRIMINANT ANALYSIS**

**-22pt27**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA

df = pd.read\_excel('/content/Linear Discriminant Analysis.xls')

x=df[['Experience','Salary']]

y=df['Attrition']

le = LabelEncoder()

y = le.fit\_transform(y)

print(x,y)

**OUPUT**

**Experience Salary**

**0 2 3.0**

**1 3 8.5**

**2 3 7.5**

**3 4 9.0**

**4 4 12.0**

**5 4 11.0**

**6 5 16.0**

**7 6 15.0**

**8 7 21.0 [1 0 0 1 0 0 0 1 0]**

 clf = LDA()

 clf.fit(x, y)

for experience, salary in zip(x['Experience'], x['Salary']):

    input\_df = pd.DataFrame([[experience, salary]], columns=['Experience', 'Salary'])

    prediction = clf.predict(input\_df)

    print(f"Experience: {experience}, Salary: {salary}, Prediction: {prediction}")

**OUTPUT**

**Experience: 2, Salary: 3.0, Prediction: [1]**

**Experience: 3, Salary: 8.5, Prediction: [0]**

**Experience: 3, Salary: 7.5, Prediction: [0]**

**Experience: 4, Salary: 9.0, Prediction: [1]**

**Experience: 4, Salary: 12.0, Prediction: [0]**

**Experience: 4, Salary: 11.0, Prediction: [0]**

**Experience: 5, Salary: 16.0, Prediction: [0]**

**Experience: 6, Salary: 15.0, Prediction: [1]**

**Experience: 7, Salary: 21.0, Prediction: [0]**

# Create a scatter plot

plt.figure(figsize=(10, 6))

# Plot the data points with different colors for different classes

for class\_value in np.unique(y):

    row\_ix = np.where(y == class\_value)[0]

    plt.scatter(x.iloc[row\_ix, 0], x.iloc[row\_ix, 1], label=f'Class {class\_value}')

# Create a mesh to plot the decision boundary

x\_min, x\_max = x['Experience'].min() - 1, x['Experience'].max() + 1

y\_min, y\_max = x['Salary'].min() - 1, x['Salary'].max() + 1

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.01), np.arange(y\_min, y\_max, 0.01))

grid = np.c\_[xx.ravel(), yy.ravel()]

predictions = clf.predict(grid).reshape(xx.shape)

# Plot the decision boundary

plt.contourf(xx, yy, predictions, alpha=0.3, cmap=plt.cm.Paired)

plt.xlabel('Experience')

plt.ylabel('Salary')

plt.title('Scatter Plot with LDA Decision Boundary')

plt.legend()

plt.show()

**OUTPUT**

