**NFT ASSIGNMENT**

**BT21CSE027: SHRUTI BADJATE**

**BT21CSE061: KANISHKA**

**Aim:** The aim of this task is to predict the future urban growth by simulating the consecutive land-use land-cover (LULC) maps of previous years in the given datasets. Dataset Proximity to CBD

**Given Data:**

LULC maps (1991, 2001, 2011 and 2021)

Proximity to roads

Proximity to CBD

It is known that areas near the roads and CBD will show major urban growth and a high

value of elevation will show a lower growth. The LULC classes are of four types: Urban,

Vegetation, Other and Water.

**Output Required**: Transition rules for projecting LULC from the year 2021 to 2031.

**Code:**

import rasterio

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import Perceptron

from sklearn.metrics import mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

# Define the file paths in the "Downloads" folder

lulc\_path1 = r"C:\Users\kanis\Downloads\Actual\_1994.tif"

lulc\_path2 = r"C:\Users\kanis\Downloads\Actual\_1999.tif"

cbd\_path = r"C:\Users\kanis\Downloads\cbddist.tif"

# Load the first LULC map

with rasterio.open(lulc\_path1) as lulc\_file1:

    lulc\_data1 = lulc\_file1.read(1)

# Load the second LULC map

with rasterio.open(lulc\_path2) as lulc\_file2:

    lulc\_data2 = lulc\_file2.read(1)

# Load the CBD map

with rasterio.open(cbd\_path) as cbd\_file:

    cbd\_data = cbd\_file.read(1)

# Preprocess data (normalize LULC data)

lulc\_data1 = (lulc\_data1 - lulc\_data1.min()) / (lulc\_data1.max() - lulc\_data1.min())

lulc\_data2 = (lulc\_data2 - lulc\_data2.min()) / (lulc\_data2.max() - lulc\_data2.min())

# Concatenate the two LULC maps to create input features

X = np.stack((lulc\_data1, lulc\_data2), axis=-1)

# Flatten the input features

X = X.reshape(-1, 2)  # Flatten the input features

y = cbd\_data.reshape(-1, 1)

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create and train the Madaline model

madaline\_model = Perceptron()

madaline\_model.fit(X\_train, y\_train.ravel())

y\_pred = madaline\_model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

accuracy = r2\_score(y\_test, y\_pred)  # Calculate R-squared (accuracy) instead of MSE

print(f"Mean Squared Error: {mse}")

print(f"Accuracy : {accuracy:.2f}")

# Visualize some predictions

sample\_indices = np.random.choice(X\_test.shape[0], 10, replace=False)

sample\_X = X\_test[sample\_indices]

sample\_y\_pred = madaline\_model.predict(sample\_X)

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)

plt.title("Actual CBD Proximity")

plt.imshow(y\_test[sample\_indices].reshape(10, -1), cmap='viridis')

plt.colorbar()

plt.subplot(1, 2, 2)

plt.title("Predicted CBD Proximity")

plt.imshow(sample\_y\_pred.reshape(10, -1), cmap='viridis')

plt.colorbar()

plt.show()

**Dataset:**



**Output:**

