***Source Code***

1.Install required libraries

!pip install pandas numpy scikit-learn matplotlib pillow scikit-image gradio

2.Upload Datasetfrom

google.colab import files

import pandas as pd

uploaded = files.upload()   # choose colors.csv file

df = pd.read\_csv("colors.csv")

df.head()

3.Data Exploration

df.info()

print("Shape:", df.shape)

df.sample(10)

df.describe(include='all')

4.Check Missing Values & Duplicates

print("Missing values:\n", df.isnull().sum())

print("Duplicates:", df.duplicated().sum())

df = df.drop\_duplicates()

5.Detect RGB Columns (or Convert HEX)

import numpy as np

numeric\_cols = df.select\_dtypes(include=[np.number]).columns.tolist()

print("Numeric columns:", numeric\_cols)

# Assume last 3 numeric columns are RGB

if len(numeric\_cols) >= 3:

    rgb\_cols = numeric\_cols[-3:]

else:

    # If dataset has HEX column instead

    def hex\_to\_rgb(h):

        h = h.lstrip('#')

        return tuple(int(h[i:i+2],16) for i in (0,2,4))

    df[['R','G','B']] = df['hex'].apply(lambda x: pd.Series(hex\_to\_rgb(x)))

    rgb\_cols = ['R','G','B']

print("RGB columns used:", rgb\_cols)

6.Visualize a Few Colors

import matplotlib.pyplot as plt

def show\_color(rgb, label=""):

    plt.imshow(np.ones((20,60,3),dtype='uint8')\*np.array(rgb,dtype='uint8'))

    plt.axis('off')

    plt.title(label)

    plt.show()

for i in range(5):

    row = df.iloc[i]

    rgb = row[rgb\_cols].tolist()

    show\_color(rgb, label=f"Row {i}")

7.Convert RGB to LAB (for perceptual distance)

from skimage.color import rgb2lab

rgb\_arr = df[rgb\_cols].to\_numpy().astype(float)/255.0

lab\_arr = rgb2lab(rgb\_arr.reshape(-1,1,3)).reshape(-1,3)

8.Nearest Neighbor Model

from sklearn.neighbors import NearestNeighbors

import joblib

nn = NearestNeighbors(n\_neighbors=1, metric='euclidean').fit(lab\_arr)

# Choose name column (adjust if needed)

name\_col = df.columns[0]

joblib.dump({'df':df, 'lab':lab\_arr, 'nn':nn, 'rgb\_cols':rgb\_cols, 'name\_col':name\_col}, "nn\_lookup.joblib")

9.Predict Color Name from RGB

from skimage.color import rgb2lab

def rgb\_to\_lab(rgb):

    arr = np.array(rgb).reshape(1,1,3)/255.0

    return rgb2lab(arr).reshape(3)

def predict\_color(rgb):

    lab = rgb\_to\_lab(rgb).reshape(1,-1)

    dist, idx = nn.kneighbors(lab)

    i = int(idx[0][0])

    name = df.iloc[i][name\_col]

    return name, df.iloc[i][rgb\_cols].tolist(), float(dist[0][0])

# Test

print(predict\_color((255,0,0)))  # Red

print(predict\_color((0,255,0)))  # Green

print(predict\_color((0,0,255)))  # Blue

10.Extract Dominant Color from Image

from PIL import Image

from sklearn.cluster import KMeans

def extract\_dominant\_rgb(image, k=3, resize=150):

    img = image.resize((resize, resize)).convert('RGB')

    arr = np.array(img).reshape(-1,3)

    kmeans = KMeans(n\_clusters=k, random\_state=42).fit(arr)

    counts = np.bincount(kmeans.labels\_)

    dominant = kmeans.cluster\_centers\_[np.argmax(counts)]

    return tuple(map(int, dominant))

11.Gradio Interface

import gradio as gr

def predict\_from\_image(img):

    image = Image.fromarray(img)

    rgb = extract\_dominant\_rgb(image)

    name, nearest\_rgb, dist = predict\_color(rgb)

    return f"Predicted Name: {name}\nInput RGB: {rgb}\nNearest Match: {nearest\_rgb}, Distance: {dist}"

iface = gr.Interface(

    fn=predict\_from\_image,

    inputs=gr.Image(type="numpy"),

    outputs="text",

    title="Naan Mudhulvan Color Detection"

)

iface.launch()