**Project Overview**

This project focuses on **predicting skin cancer** using the **HAM10000 dataset**, which contains images of skin lesions along. We apply various data preprocessing techniques such as missing data handling, categorical encoding, feature engineering, and outlier removal, followed by exploratory data analysis to extract insights for better classification of skin cancer.

**Group Members & Roles**

| **IT Number** | **Name** | **preprocessing technique** |
| --- | --- | --- |
| IT23709720 | Herath H.M.G.S.B | Normalization |
| IT24100382 | Herath H.M.P.C.B | Handling missing data |
| IT24100777 | Amarasinghe H.M.S.L |  |
| IT24102254 | Rajapaksha R.P.H.L | LDA |
| IT24101858 | Gunathilaka D.T.G.N.M | Encoding categorical variables |
| IT24101183 | Dharmasiri W.K.Y.D | Data augmentation |

**Dataset Information**

**Primary Dataset**

* **Name**: HAM10000: A Large Collection of Skin Cancer Images
* **Source**: https://www.kaggle.com/datasets/farjanakabirsamanta/skin-cancer-dataset
* **File Format**: JPG (Images)
* **Description**: The dataset includes skin lesion images

**1. Setup and Install Dependencies**

1. **Install required libraries** from requirements.txt:
2. pip install -r requirements.txt

**2. Mount Google Drive in Colab**

from google.colab import drive

drive.mount('/content/drive')

**3. Dataset Loading**

Each member should load the dataset before applying their preprocessing steps:

import pandas as pd

df = pd.read\_csv('/content/drive/MyDrive/Skin\_Cancer\_HAM10000/archive/HAM10000\_metadata.csv')

df.head()

**4. Individual Tasks (Each Member)**

**Herath H.M.G.S.B - Normalization**

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df['age'] = scaler.fit\_transform(df[['age']])

df.head()

**Herath H.M.P.C.B - Handling Missing Data**

df['age'].fillna(df['age'].mean(), inplace=True)

df.head()

**Amarasinghe H.M.S.L - Outlier Removal**

Q1 = df['age'].quantile(0.25)

Q3 = df['age'].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

df\_no\_outliers = df[(df['age'] >= lower\_bound) & (df['age'] <= upper\_bound)]

df\_no\_outliers.head()

**Rajapaksha R.P.H.L - LDA**

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

lda = LinearDiscriminantAnalysis(n\_components=2)

X\_lda = lda.fit\_transform(df[['age', 'sex']], df['dx'])

X\_lda[:5]

**Gunathilaka D.T.G.N.M - Encoding Categorical Variables**

from sklearn.preprocessing import LabelEncoder

label\_encoder = LabelEncoder()

df['sex'] = label\_encoder.fit\_transform(df['sex'])

df = pd.get\_dummies(df, columns=['localization'], drop\_first=True)

df.head()

**Dharmasiri W.K.Y.D - Feature Engineering**

df['age\_group'] = pd.cut(df['age'], bins=[0, 30, 60, 100], labels=['Young', 'Middle-Aged', 'Old'])

df.head()

**5. Final Pipeline Execution**

After each member completes their task, integrate and run the final pipeline:

jupyter notebook notebooks/group\_pipeline.ipynb