Uncover Patterns ring in Dermatological Data

Dermatology

DESCRIPTIVE ANALYSIS

GROUP 04:

NAETHREE(15552)

DINULA(15369)

ASLAM(15338)

Abstract

This report presents a comprehensive analysis of a dermatology dataset, which encompasses various clinical and histopathological features related to skin conditions. The dataset includes a range of ordinal variables, reflecting the severity or presence of symptoms, as well as 'age' variable. This analysis focuses on basic descriptive statistics, visualization of feature relationships, and techniques to explore associations among variables. The insights gained from this analysis aim to lay the groundwork for subsequent efforts in classifying erythemato-squamous diseases, utilizing the identified patterns and associations among the features.

Table of Contents

List of Figures

List of Tables

1. Introduction

The differential diagnosis of erythemato-squamous diseases is recognized as a challenging problem in dermatology due to the shared clinical features of erythema and scaling [2]. The disorders within this group include psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris [2]. Accurate diagnosis is crucial for effective treatment but is often dependent on biopsy, which can be invasive and inconvenient for patients. Additionally, the histopathological similarities among these conditions complicate the diagnostic process. Promising solutions for improving diagnostic accuracy and efficiency are offered by advancements in machine learning and artificial intelligence. This report lays the foundation for the development of a robust model by exploring key relationships through EDA.

2. Description of the Question

Using advanced analytical techniques to enhance classification accuracy is essential due to the limitations of traditional diagnostic methods. Therefore, the objective of this report is to perform exploratory data analysis (EDA) to address the following questions:

1) What are the factors mainly affecting each of the six diseases?

2) How can we accurately classify erythemato-squamous diseases based on clinical and histopathological features?

In pursuit of these objectives, various statistical methods will be employed to uncover the underlying patterns and relationships within the dataset. By analyzing both clinical and histopathological features, this report aims to identify key factors that differentiate each of the six diseases.

3. Description of the Dataset

The dataset contains 35 variables and 366 records.
Few of the variables are given below:

Name of variable	Description of variable	Type of variable
erythema	Redness of the skin caused by increased blood flow	Ordinal (0,1,2,3)
scaling	Flaking or shedding of the outer layer of the skin	Ordinal (0,1,2,3)
definite_borders	Clear and distinct edges of a lesion or affected area	Ordinal (0,1,2,3)
itching	Sensation causing the desire to scratch	Ordinal (0,1,2,3)
koebner_phenomenon	Appearance of new skin lesions on previously unaffected skin due to trauma	Ordinal (0,1,2,3)
polygonal_papules	Small, raised, polygonal-shaped bumps on the skin	Ordinal (0,1,2,3)
follicular_papules	Small, raised bumps centered around a hair follicle.	Ordinal (0,1,2,3)
age	Age of the patient	Continuous

family_history	Indicates if any of the diseases are present in the family (1 if yes, 0 if no).	Nominal(0,1)
class	The target variable indicates the disease class. 1-Psoriasis, 2-Seborrheic Dermatitis, 3-Lichen Planus, 4-Pityriasis Rosea, 5-Chronic Dermatitis, 6-Pityriasis Rubra Pilaris	Nominal

Table 1: Description of part of the dataset

The rest of the variables are all ordinal each with 0-3 symptom levels and they are oral_mucosal_involvement, knee_and_elbow_involvement, scalp_involvement, melanin_incontinence, eosinophils_infiltrate, PNL_infiltrate, fibrosis_papillary_dermis, exocytosis, acanthosis, hyperkeratosis, parakeratosis, clubbing_rete_ridges, elongation_rete_ridges, thinning_suprapapillary_epidermis, spongiform_pustule, munro_microabcess, focal_hypergranulosis, disappearance_granular_layer, vacuolisation_damage_basal_layer, spongiosis, saw_tooth_appearance_retes, follicular_horn_plug, perifollicular_parakeratosis, inflammatory_mononuclear_infiltrate, band_like_infiltrate.

Figure 1: Response variable Diseases

4. Pre-Processing

- The datatype of 'age' was changed from object to int64.
- Datatype of all other features were changed from int64 to category for plotting purposes.
- 8 missing values(denoted as '?') were found in the 'age' variable. Since the evidences from the Shapiro Wilk test(p value:0.002 <0.05) and the Q-Q plot confirmed the distribution of 'age' to be not normal, median Imputation (Median age=35) was done instead of mean imputation.
- No duplicates were found.
- For further multivariate analysis ordinal variables were converted to numeric.

Figure 2: Q-Q Plot of Age

5. Main results of the Descriptive Analysis

5.1 Univariate Analysis

5.1.1 Response Variable

Figure 3 denotes percentage bar chart of the response and it is evident that the highest percentage of data is available for Psoriasis while Pityriasis Rubra Pilaris is indicated by the least amount of data.

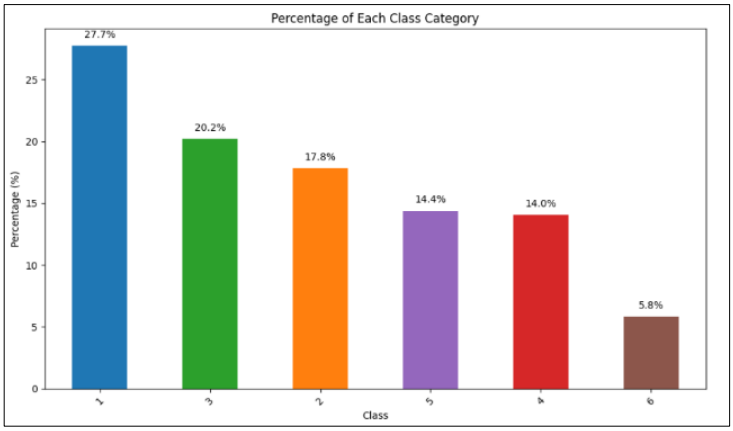

Figure 3: Percentage Bar Chart of Class

5.1.2 Age variable

Age's distribution can be seen by Figure 4.The mean age of patients is 35.7 while less patients are aged greater than 60, hence this should be kept in mind when reflecting the results of the dataset as it might not represent patients with extreme ages well.

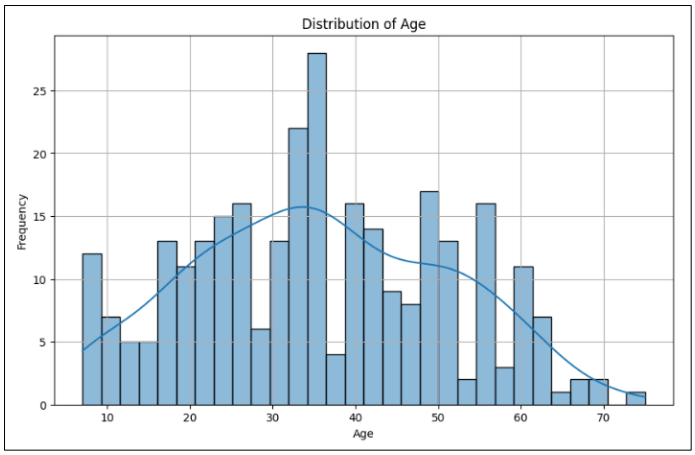

Figure 4: Distribution of age

5.1.2 Nominal variable family history

By Figure 5, it is evident that most patients are from families who don't have any of these diseases. When building and evaluating models, it's essential to keep this in mind since the results won't represent the patients from families with the diseases well.

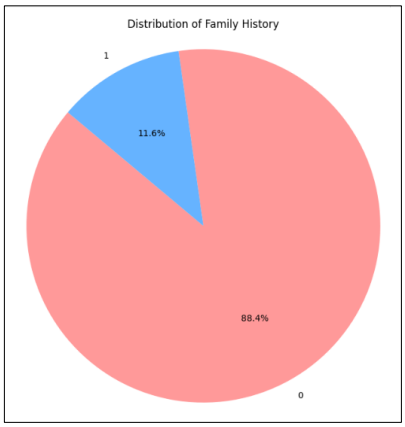

Figure 5: Distribution of family history

5.2 Bivariate Analysis

5.2.1 Main observations of variables by Class

According to Figure 6, most people affected by Pityriasis Rosea(class 4) seem to have no itching. This is confirmed by "***Except for mild to severe itching in up to 25% of patients, no systemic symptoms are typically present during the rash phase of pityriasis rosea.***"**[1]** Majority of people affected by Lichen Planus have severe itching which is confirmed by "***Lichen planus is a non-infectious, itchy rash that can affect many areas of the body.***"**[2]**

According to Figure 7, people affected by Psoriasis, Seborrheic Dermatitis, Pityriasis Rosea, Chronic Dermatitis and Pityriasis Rubra Pilaris aren't affected by the oral mucosa, but Lichen Planus commonly has oral mucosal involvement where patients face them at different severity levels according to figure 7. This is confirmed by ***"Oral lichen planus is often diagnosed by a dentist during routine dental check-ups."***[2] which indicates that oral lichen planus is a type of Lichen Planus, which affects the mouth.

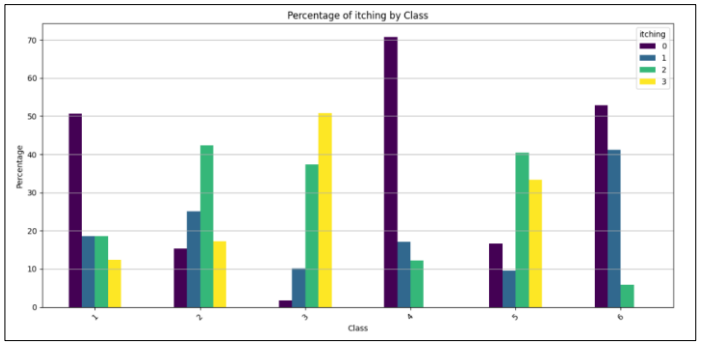

Figure 6: Percentage bar chart of itching by class

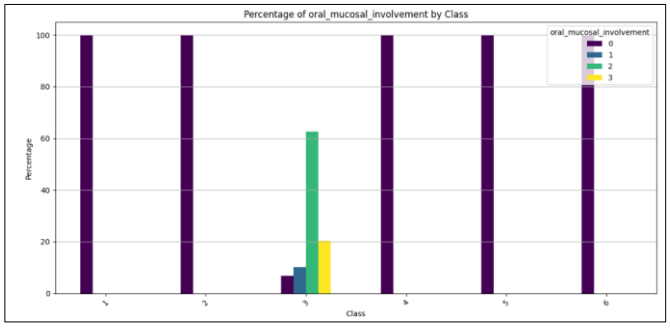

Figure 7: Percentage bar chart of oral mucosal involvement by class

Based on research[3]-[11] regarding the six diseases, the below Table 2 shows the following:
Green: the symptoms reflected in the dataset and the references
Turquoise: the symptoms represented by the dataset and not the references
No color: the symptoms in the references but not in the dataset

Psoriasis	Seborrheic Dermatitis	Lichen Planus	Pityriasis Rosea	Chronic Dermatitis	Pityriasis Rubra Pilaris
Erythema	Erythema	Erythema	Erythema	Erythema	Erythema
Scaling	Scaling	Definite Borders	Scaling	Scaling	Scaling
Definite Borders	Itching	Itching	Definite Borders	Itching	Definite Borders
Itching	Scalp Involvement	Polygonal Papules	Itching	Spongiosis	Follicular Papules
Koebner Phenomenon	Exocytosis	Oral Mucosal Involvement	Exocytosis	Scalp Involvement	Knee and Elbow Involvement
Knee and Elbow Involvement	Acanthosis	Melanin Incontinence	Spongiosis	Fibrosis Papillary Dermis	Scalp Involvement
Scalp Involvement	Parakeratosis	Exocytosis	Inflammatory Mononuclear Infiltrate	Inflammatory Mononuclear Infiltrate	Hyperkeratosis
PNL Infiltrate	Spongiosis	Hyperkeratosis	Koebner Phenomenon	Definite Borders	Parakeratosis
Acanthosis	Inflammatory Mononuclear Infiltrate	Focal Hypergranulosis	Acanthosis	Exocytosis	Inflammatory Mononuclear Infiltrate
Parakeratosis	PNL Infiltrate	Vacuolisation Damage Basal Layer	Parakeratosis	Acanthosis	Exocytosis
Clubbing Rete Ridges	Definite Borders	Saw Tooth Appearance Retes		Elongation Rete Ridges	Acanthosis
Elongation Rete Ridges		Inflammatory Mononuclear Infiltrate			Spongiosis
Thinning Suprapapillary Epidermis		Band-like Infiltrate			Follicular Horn Plug
Spongiform Pustule		Scaling			Perrifollicular Parakeratosis
Munro Microabcess		Koebner Phenomenon			

Disappearance Granular Layer		Acanthosis			
Inflammatory Mononuclear Infiltrate		Parakeratosis			
		Spongiosis			

Table 2: Comparison of symptoms for each class of diseases by research VS analysis

6.2.2 Correlation between the numeric variables

By Figure 8, the correlation coefficient values between the numeric variables can be observed.
Key observations are as follows:

- Each pair in Focal hypergranulosis, Polygonal Papules, Oral Mucosal Involvement, Melanin Incontinence, Vascuolisation Damage Basal Layer, Saw Tooth Appearance Retes and Band Like Infiltrate is highly correlated with each other. These features maybe highly correlated because they are commonly found together in certain dermatological conditions, particularly lichen planus.
- Each pair in Follicular Papules, Follicular Horn Plug and Perifollicular Parakeratosis are highly correlated with each other.

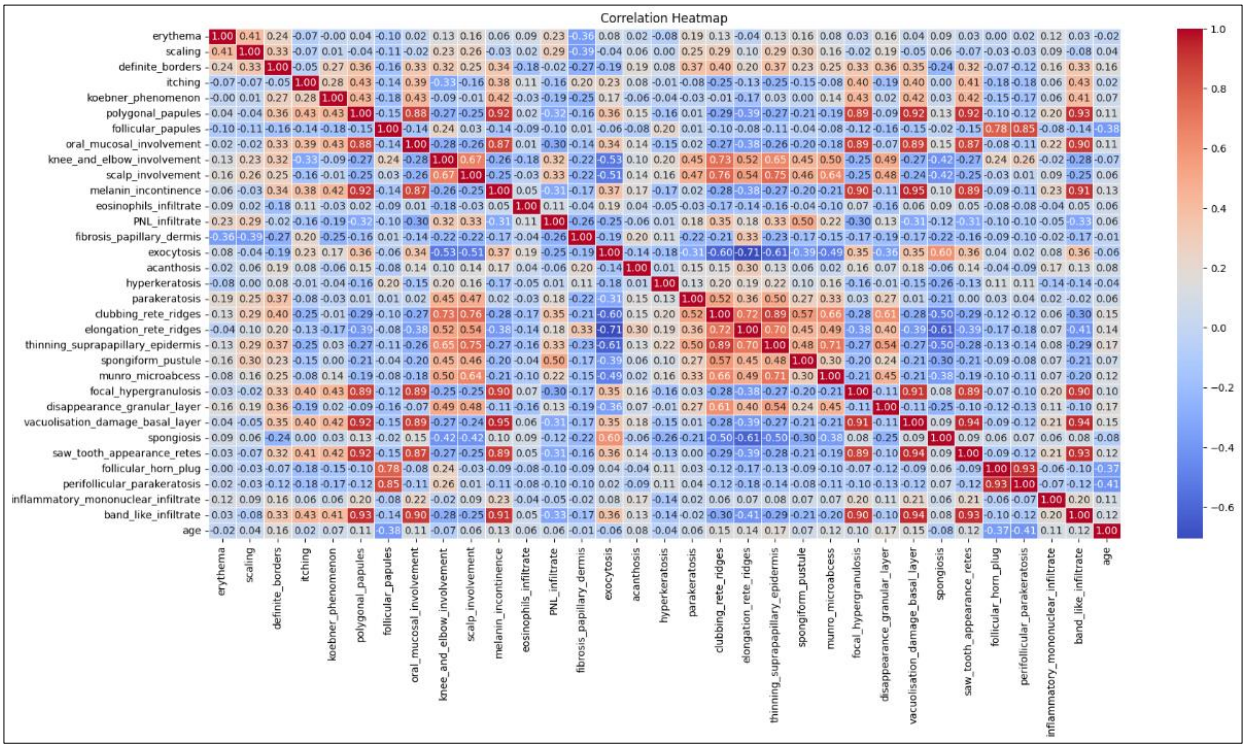

Figure 8: Correlation Heatmap of numerical variables

5.3 Multivariate Analysis

5.3.1 Multicollinearity

The Variance Inflation Factor analysis was conducted to assess the multicollinearity among the features in the dermatology dataset. The results in Table 3 indicate significant multicollinearity, particularly among certain

features. For instance, features like vacuolisation_damage_basal_layer (VIF = 27.79), band_like_infiltrate (VIF = 17.76), and melanin_incontinence (VIF = 16.30) exhibit very high VIF scores, suggesting a high degree of correlation with other predictors in the model. Other features such as polygonal_papules, erythema, and scaling also show substantial multicollinearity with VIF scores above 10.

Variable Name	VIF Score	Variable Name	VIF Score
erythma	13.840555	hyperkeratosis	1.921288
scaling	12.067053	parakeratosis	5.140062
definite_borders	8.825341	clubbing_rete_ridges	12.034817
itching	4.30289	elongation_rete_ridges	11.559817
koebner_phenomenon	2.398118	thinning_suprapapillary_epidermis	10.070708
polygonal_papules	13.57255	spongiform_pustule	2.419026
follicular_papules	4.412315	munro_microabcess	3.0005
oral_mucosal_involvement	9.001588	focal_hypergranulosis	10.176414
knee_and_elbow_involvement	4.903507	disappearance_granular_layer	2.468943
scalp_involvement	4.53806	vacuolisation_damage_basal_layer	27.794759
family_history	1.571547	spongiosis	4.604313
melanin_incontinence	16.303299	saw_tooth_appearance_retes	15.529731
eosinophils_infiltrate	1.353087	follicular_horn_plug	8.397643
PNL_infiltrate	2.724906	perifollicular_parakeratosis	12.400189
fibrosis_papillary_dermis	3.778801	inflammatory_mononuclear_infiltrate	9.527857
exocytosis	6.957575	band_like_infiltrate	17.755538
acanthosis	10.79972	age	7.202199

Table 3: Table of VIF values

5.3.2 Principal Component Analysis

5.3.2.1 Analysis of Loadings Plot

Figure 9, PCA was applied on the numeric features(ordinal variables were converted to numeric), and the loadings plot was visualized. This illustrates the contribution of each feature to the first two principal components. However, it should be noted that the first two principal components account for only 40% of the total variance in the data. This relatively low percentage of explained variance suggests that a significant portion of the dataset's information is not captured by these two components. Consequently, the results derived from this analysis should be interpreted with caution, as they may not fully represent the underlying structure of the data and could lead to potentially invalid conclusions.

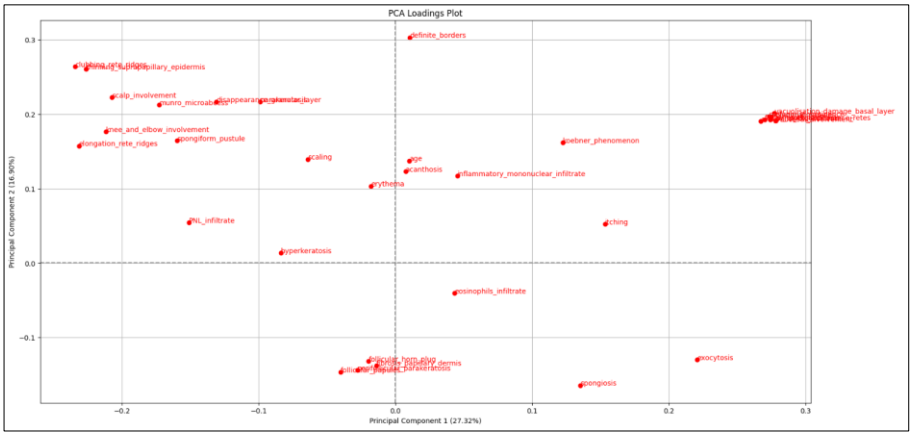

Figure 9: PCA Loadings plot

5.3.2.2 Detection of Clusters

By Figure 10, it is evident that 3 clusters were identified following the application of PCA to the dataset. However, it should be noted that the first two principal components, which were used to determine these clusters, only account for a limited proportion of the total variance in the data. Hence the identified clusters might not represent the true data distribution accurately.

However, the clusters were analyzed to observe any patterns. By Figure 11, the Cluster 2 (blue) represents Class 1(Psoriasis) while the Cluster 0 (green) represents Class 3(Lichen Planus). The Cluster 1 seems to have a mix of classes 2,4,5 and 6. (Seborrheic Dermatitis, Pityriasis Rosea, Chronic Dermatitis, Pityriasis Rubra Pilaris).

To understand the clusters better and to identify the features that might be contributing to the classes, the Figures 12,13 and 14 were sketched.

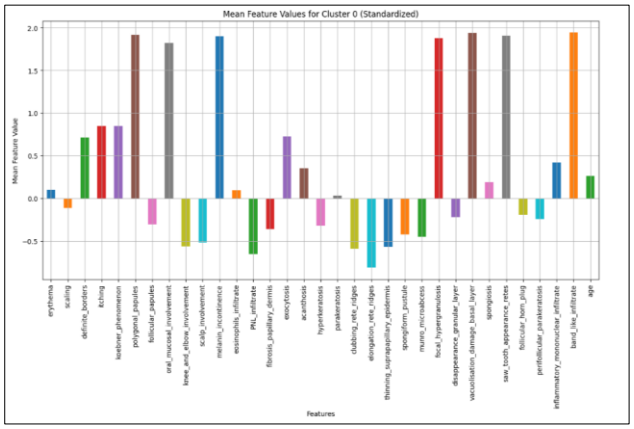

Figure 12: Mean Features values for Cluster 0

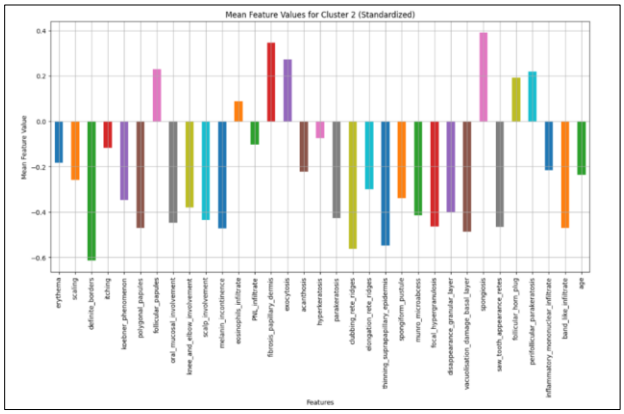

Figure 14: Mean Features values for Cluster 2

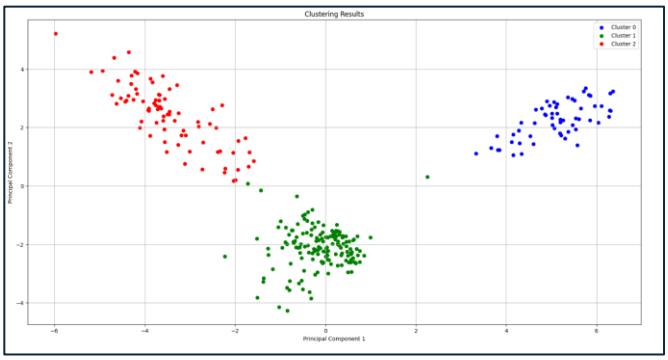

Figure 10: Clusters after PCA

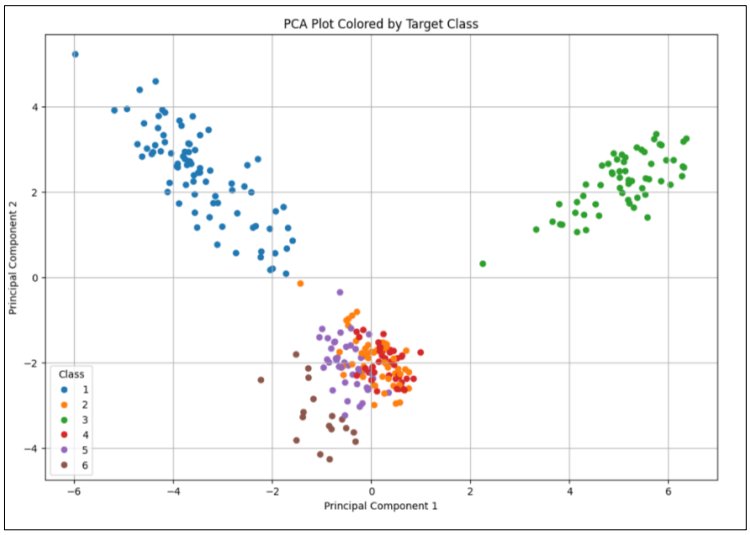

Figure 11: Clusters by Class

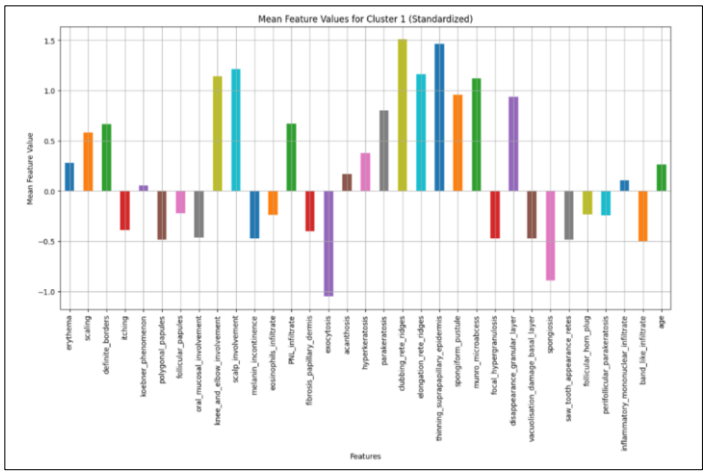

Figure 13: Mean Features values for Cluster 1

The mean values of features for clusters 0, 1, and 2 were calculated in Figures 12, 13 and 14. However, the results may not be entirely accurate since the principal components explain only a small portion of the variance.

5.4 Outlier Detection

In Figure 15, Mahalanobis Distance with robust covariance estimation was employed to detect 101 outliers. The threshold for identifying outliers was derived from the chi-squared distribution, also set to a 1% significance level.

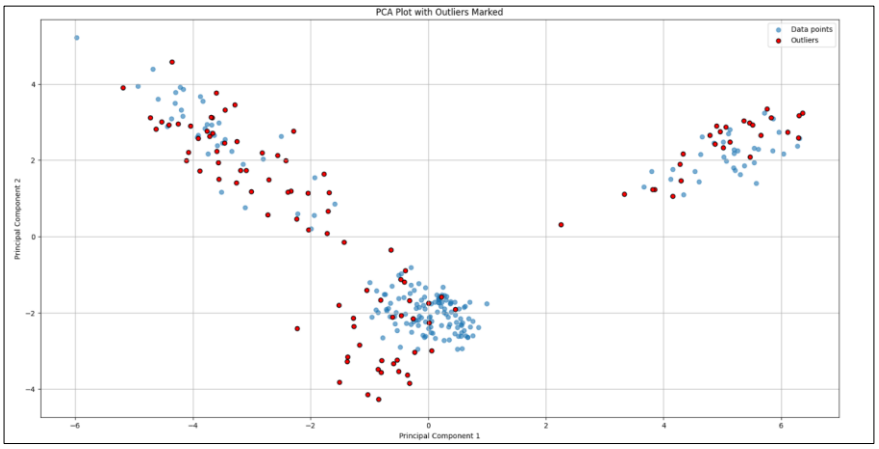

Figure 15: PCA plot with outliers marked(Mahalanobis)

In Figure 16, Isolation Forest, an ensemble-based anomaly detection technique, detected 3 outliers by isolating data points through recursive partitioning. This method identified a set of outliers based on the contamination parameter, which was set to 1% of the data.

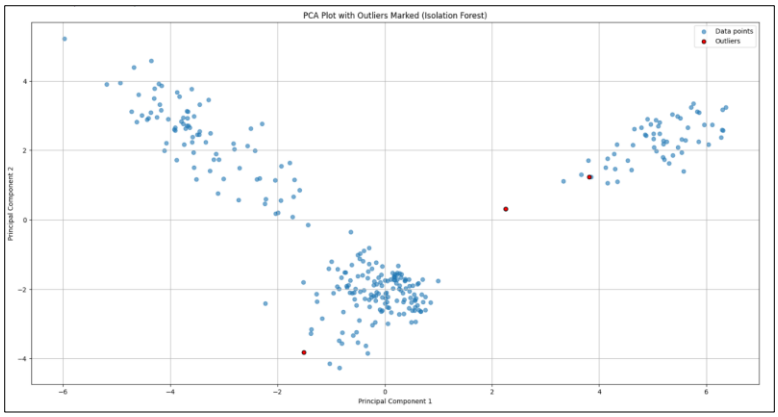

Figure 16: PCA plot with outliers marked(Isolation Forest)

6. Suggestions for a quality advanced analysis

1) Models will be developed both with and without outliers to assess their impact on performance. When deciding whether to remove outliers, careful consideration will be given to minimize the loss of valuable information. By comparing the results from both scenarios, we can determine the optimal strategy for handling outliers in this dataset.

2) Since few variables are highly correlated and since multicollinearity exists, few of them could be removed after careful consideration of supporting researches. Also models that handle multicollinearity can be used for advanced analysis. After building the model, feature selection techniques like Feature importance in Random Forest, or SHAP values can be used.

3) Classes 4, 5, and 6 have relatively fewer observations, with percentages of 14.0%, 14.4%, and 5.8%, respectively, indicating a significant class imbalance. In contrast, classes 1 and 2 have a higher number of observations, comprising 27.7% and 20.2% of the dataset, respectively. This imbalance can potentially affect the performance of classification models by biasing them towards the more frequent classes. To address this issue and ensure balanced performance across all classes, several techniques will be employed in the advanced analysis. These include oversampling the minority classes using methods like SMOTE (Synthetic Minority Over-sampling Technique), under sampling the majority classes, or utilizing algorithms that are robust to class imbalance, such as balanced random forests.

7. References

[1] DermNet. "Pityriasis Rosea," DermNet NZ. [Online]. Available: https://dermnetnz.org/topics/pityriasis-rosea. [Accessed: 20-Jul-2024].

[2] NHS Inform. "Lichen Planus," NHS Inform. [Online]. Available: https://www.nhsinform.scot/illnesses-and-conditions/skin-hair-and-nails/lichen-planus/. [Accessed: 22-Jul-2024].

[3] Mayo Clinic. "Lichen Planus - Symptoms and Causes," [Online]. Available: https://www.mayoclinic.org/diseases-conditions/lichen-planus/symptoms-causes/syc-20351378. [Accessed: 23-Jul-2024].

[4] National Institute of Arthritis and Musculoskeletal and Skin Diseases (NIAMS). "Psoriasis - Symptoms and Causes," [Online]. Available: https://www.niams.nih.gov/health-topics/psoriasis#:~:text=Symptoms%20of%20psoriasis%20vary%20from,Thick%2C%20ridged%2C%20pitted%20nails. [Accessed: 22-Jul-2024].

[5] NHS. "Psoriasis - Symptoms," [Online]. Available: https://www.nhs.uk/conditions/psoriasis/symptoms/. [Accessed: 23-Jul-2024].

[6] National Eczema Association. "Types of Eczema: Seborrheic Dermatitis," [Online]. Available: https://nationaleczema.org/eczema/types-of-eczema/seborrheic-dermatitis/. [Accessed: 23-Jul-2024].

[7] Mayo Clinic. "Seborrheic Dermatitis - Symptoms and Causes," [Online]. Available: https://www.mayoclinic.org/diseases-conditions/seborrheic-dermatitis/symptoms-causes/syc-20352710. [Accessed: 24-Jul-2024].

[8] NHS. "Pityriasis Rosea - Symptoms," [Online]. Available: https://www.nhs.uk/conditions/pityriasis-rosea/#:~:text=A%20widespread%20rash%20of%20small,body%20and%20may%20be%20itchy. [Accessed: 25-Jul-2024].

[9] Mayo Clinic. "Dermatitis - Symptoms and Causes," [Online]. Available: https://www.mayoclinic.org/diseases-conditions/dermatitis-eczema/symptoms-causes/syc-20352380#:~:text=Dermatitis%20is%20a%20common%20condition,%2C%20ooze%2C%20crust%20or%20flake. [Accessed: 23-Jul-2024].

[10] Mount Sinai. "Pityriasis Rubra Pilaris - Overview," [Online]. Available: https://www.mountsinai.org/health-library/diseases-conditions/pityriasis-rubra-pilaris#:~:text=Pityriasis%20rubra%20pilaris%20is%20an,a%20characteristic%20orange%2Dred%20color. [Accessed: 24-Jul-2024].

[11] National Organization for Rare Disorders (NORD). "Pityriasis Rubra Pilaris," [Online]. Available: https://rarediseases.org/rare-diseases/pityriasis-rubra-pilaris/. [Accessed: 22-Jul-2024].

8. Appendix including code

1

```
import warnings
warnings.filterwarnings('ignore')
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from scipy.stats import chi2
from sklearn.covariance import MinCovDet
from sklearn.ensemble import IsolationForest
```

2

```
from google.colab import drive
drive.mount('/content/drive')

file_path = '/content/drive/My Drive/ML 2/Dermatology/dermatology_database_1.csv'
```

```
Mounted at /content/drive
```

```
df = pd.read_csv(file_path)
df.head()
```

3

```
df.isna().sum()

# Function to count question marks in each column
def count_question_marks(column):
    return column.apply(lambda x: str(x).count('?')).sum()

# Applying the function to each column
question_marks_count = df.apply(count_question_marks)

print(question_marks_count)
```

```
df['age'] = pd.to_numeric(df['age'], errors='coerce')

# Check summary statistics, ignoring NaN values
mean_age = df['age'].mean()
median_age = df['age'].median()

# Print mean and median to help decide
print(f"Mean age: {mean_age}")
print(f"Median age: {median_age}")
```

```
age_data = df['age'].dropna()
import scipy.stats as stats

# Plot histogram
plt.hist(age_data, bins=10, edgecolor='black', alpha=0.7)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.show()

# Q-Q Plot
stats.probplot(age_data, dist="norm", plot=plt)
plt.title('Q-Q Plot of Age')
plt.show()

# Shapiro-Wilk Test
shapiro_test = stats.shapiro(age_data)
print(f"Shapiro-Wilk Test p-value: {shapiro_test.pvalue}")
```

```
# Replace NaN values with the median
df['age'].fillna(median_age, inplace=True)

# Convert to int64 if required
df['age'] = df['age'].astype('int64')
```

4

```
categorical_columns = [col for col in df.columns if col != 'age']

# Convert specified columns to categorical
df[categorical_columns] = df[categorical_columns].astype('category')
```

```
print(df.dtypes)
```

```
df.shape
```

```
#Check for duplicates in the entire DataFrame
duplicate_rows = df[df.duplicated()]
num_duplicates = duplicate_rows.shape[0]

print(f"Number of duplicate rows found: {num_duplicates}")
```

5

```
#Features (all columns except the target column)
X = df.drop(columns=['class'])
#Target variable
y = df['class']

#Split the dataset into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

#Print the shapes of the resulting datasets to verify the split
print("Training set - Features shape:", X_train.shape, " Target shape:", y_train.shape)
print("Testing set - Features shape:", X_test.shape, " Target shape:", y_test.shape)
```

6

```
# Define color mapping
unique_classes = [1, 2, 3, 4, 5, 6]
colors = plt.get_cmap('tab10').colors  # Get colors from 'tab10' colormap

# Ensure there are enough colors for the number of classes
color_mapping = {cls: colors[i % len(colors)] for i, cls in enumerate(unique_classes)}
```

```
# Calculate the value counts and convert to percentages
class_counts = y_train.value_counts(normalize=True) * 100

# Plotting the percentage bar chart with different colors
plt.figure(figsize=(10, 6))

# Plot the bar chart
bars = class_counts.plot(kind='bar', color=[color_mapping[cls] for cls in class_counts.index])

# Annotate each bar with its percentage value
for bar in bars.patches:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 0.5, f'{yval:.1f}%', ha='center', va='bottom', fontsize=10)

plt.xlabel('Class')
plt.ylabel('Percentage (%)')
plt.title('Percentage of Each Class Category')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

7

```
# Distribution of Age
plt.figure(figsize=(10, 6))
sns.histplot(X_train['age'], kde=True, bins=30)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.grid()
plt.show()
```

8

```
# Plot the distribution of 'family_history' in the training set
family_history_counts = X_train['family_history'].value_counts()
labels = family_history_counts.index
sizes = family_history_counts.values
colors = ['#ff9999','#66b3ff']

plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Family History')
plt.axis('equal')
plt.show()
```

9

```
# Combine X_train and y_train for easier grouping
combined_df = X_train.copy()
combined_df['class'] = y_train

# List of categorical features to plot, excluding 'age'
categorical_features = [col for col in X_train.columns if col != 'age']

# Plotting separate clustered bar charts for each categorical feature
for feature in categorical_features:
    # Calculate the percentage of each category within the groups defined by 'class'
    crosstab = pd.crosstab(combined_df['class'], combined_df[feature])
    crosstab_percentage = crosstab.div(crosstab.sum(axis=1), axis=0) * 100  # Normalize by rows

    # Plot the clustered bar chart
    ax = crosstab_percentage.plot(kind='bar', figsize=(12, 6), colormap='viridis')
    plt.xlabel('Class')
    plt.ylabel('Percentage')
    plt.title(f'Percentage of {feature} by Class')
    plt.xticks(rotation=45)
    plt.legend(title=feature)
    plt.grid(axis='y')

    # Display the plot
    plt.tight_layout()
    plt.show()
```

10

```
# Define color mapping with integers
color_mapping = {
    '1': (0.12156862745098039, 0.4666666666666667, 0.7058823529411765),
    '2': (1.0, 0.4980392156862745, 0.054901960784313725),
    '3': (0.17254901960784313, 0.6274509803921569, 0.17254901960784313),
    '4': (0.8392156862745098, 0.15294117647058825, 0.156862745098039),
    '5': (0.5803921568627451, 0.403921568627451, 0.7411764705882353),
    '6': (0.5490196078431373, 0.33725490196078434, 0.29411764705882354)
}

# Create the box plot with the consistent color mapping
plt.figure(figsize=(10, 6))
sns.boxplot(x='class', y='age', data=combined_df, palette=color_mapping)
plt.xlabel('Class')
plt.ylabel('Age')
plt.title('Box Plot of Age by Class')
plt.grid(axis='y')
plt.show()
```

11

```
# Convert ordinal features to numeric
ordinal_features = X_train.columns.difference(['family_history'])
X_train[ordinal_features] = X_train[ordinal_features].apply(pd.to_numeric)

X_train_numeric = X_train.drop(columns=['family_history'])
correlation_matrix = X_train_numeric.corr()

# Plot the heatmap
plt.figure(figsize=(20, 9))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```

12

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Convert ordinal features to numeric
ordinal_features = X_train.columns.difference(['family_history'])
X_train[ordinal_features] = X_train[ordinal_features].apply(pd.to_numeric)

# Calculate VIF for each feature
vif_data = pd.DataFrame()
vif_data['feature'] = X_train.columns
vif_data['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]

print(vif_data)
```

13

```
# Remove 'family_history' feature from both training and testing sets
X_train = X_train.drop(columns=['family_history'])
X_test = X_test.drop(columns=['family_history'])

categorical_cols = X_train.select_dtypes(include=['category']).columns
X_train[categorical_cols] = X_train[categorical_cols].astype(int)
X_test[categorical_cols] = X_test[categorical_cols].astype(int)

# Standardize the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Apply PCA
pca = PCA()
X_train_pca = pca.fit_transform(X_train_scaled)

# Create a DataFrame with PCA components and feature names
loadings_df = pd.DataFrame(pca.components_.T, columns=[f'PC{i+1}' for i in range(X_train_scaled.shape[1])], index=X_train.columns)

# Plot loadings
plt.figure(figsize=(20, 10))
plt.scatter(loadings_df.iloc[:, 0], loadings_df.iloc[:, 1], color='r')

# Annotate points with feature names
for i in range(len(X_train.columns)):
    plt.text(loadings_df.iloc[i, 0], loadings_df.iloc[i, 1], X_train.columns[i], color='r')

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA Loadings Plot')
plt.grid()
plt.axhline(0, color='grey', linestyle='--')
plt.axvline(0, color='grey', linestyle='--')
plt.show()
```

14

```
# Print the explained variance ratio for the first and second principal components
explained_variance_ratio = pca.explained_variance_ratio_
print("Explained variance ratio for Principal Component 1:", explained_variance_ratio[0])
print("Explained variance ratio for Principal Component 2:", explained_variance_ratio[1])
```

```
# Apply K-Means Clustering with the optimal number of clusters (assuming optimal_k = 3 from elbow method)
optimal_k = 3
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
clusters = kmeans.fit_predict(X_train_pca)

# Add cluster labels to the PCA DataFrame
pca_df = pd.DataFrame(X_train_pca, columns=[f'PC{i+1}' for i in range(X_train_pca.shape[1])])
pca_df['Cluster'] = clusters

# Plot clusters
plt.figure(figsize=(20, 10))
colors = ['b', 'g', 'r']
for cluster in range(optimal_k):
    cluster_data = pca_df[pca_df['Cluster'] == cluster]
    plt.scatter(cluster_data['PC1'], cluster_data['PC2'], label=f'Cluster {cluster}', color=colors[cluster])

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Clustering Results')
plt.legend()
plt.grid()
plt.show()
```

15

```
# Standardize the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)

# Apply PCA
pca = PCA(n_components=2)
X_train_pca = pca.fit_transform(X_train_scaled)

# Create a DataFrame with PCA components and target labels
pca_df = pd.DataFrame(X_train_pca, columns=['PC1', 'PC2'])
pca_df['Class'] = y_train.values  # Add target labels
# Define a distinct color palette
palette = sns.color_palette('tab10', n_colors=len(y_train.unique()))
# Plot PCA results with colors representing classes
plt.figure(figsize=(12, 8))
sns.scatterplot(x='PC1', y='PC2', hue='Class', data=pca_df, palette=palette, edgecolor=None)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA Plot Colored by Target Class')
plt.legend(title='Class')
plt.grid()
plt.show()
```

16

```
#Analysis of clusters
optimal_k = 3
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
clusters = kmeans.fit_predict(X_train_pca)

# Add cluster labels to the PCA DataFrame
pca_df['Cluster'] = clusters

# Calculate the mean value of each feature for each cluster
df_with_clusters = pd.DataFrame(X_train_scaled, columns=X_train.columns)  # Use scaled data
df_with_clusters['Cluster'] = clusters
cluster_means = df_with_clusters.groupby('Cluster').mean()

# Define a color palette
colors = sns.color_palette("tab10", n_colors=len(X_train.columns))

# Plot the mean values of features for each cluster separately with distinct colors for each feature
for cluster in range(optimal_k):
    plt.figure(figsize=(15, 7))
    cluster_means.iloc[cluster].plot(kind='bar', color=colors)
    plt.title(f'Mean Feature Values for Cluster {cluster} (Standardized)')
    plt.xlabel('Features')
    plt.ylabel('Mean Feature Value')
    plt.grid()
    plt.show()

# Calculate the mean value of each feature for each class
df_with_classes = pd.DataFrame(X_train_scaled, columns=X_train.columns)  # Use scaled data
df_with_classes['Class'] = y_train.values
class_means = df_with_classes.groupby('Class').mean()

# Plot the mean values of features for each class with distinct colors for each feature
plt.figure(figsize=(15, 7))
class_means.plot(kind='bar', color=colors, figsize=(35, 7))
plt.title('Mean Feature Values for Each Class (Standardized)')
plt.xlabel('Class')
plt.ylabel('Mean Feature Value')
plt.legend(title='Features')
plt.grid()
plt.show()
```

17
`# Find outliers using Mahalanobis distance with robust covariance`
`robust_cov = MinCovDet().fit(X_train_scaled)`
`md_robust = robust_cov.mahalanobis(X_train_scaled)`
`threshold_robust = chi2.ppf((1 - 0.01), df=X_train_scaled.shape[1])`
`outliers_robust = np.where(md_robust > threshold_robust)[0]`
 	
`# Count of outliers`
`num_outliers_robust = len(outliers_robust)`
`print(f"Number of outliers (robust): {num_outliers_robust}")`
 	
`# PCA plot with marked outliers`
`plt.figure(figsize=(20, 10))`
`plt.scatter(X_train_pca[:, 0], X_train_pca[:, 1], label='Data points', alpha=0.6)`
`plt.scatter(X_train_pca[outliers_robust, 0], X_train_pca[outliers_robust, 1], color='r', label='Outliers', edgecolors='k')`
 	
`plt.xlabel('Principal Component 1')`
`plt.ylabel('Principal Component 2')`
`plt.title('PCA Plot with Outliers Marked')`
`plt.legend()`
`plt.grid()`
`plt.show()`
	18
	`# Use Isolation Forest to find outliers`
	`iso_forest = IsolationForest(contamination=0.01, random_state=42)`
	`outlier_labels = iso_forest.fit_predict(X_train_scaled)`
	`outliers_iso = np.where(outlier_labels == -1)[0]`
	
	`# Count of outliers using Isolation Forest`
	`num_outliers_iso = len(outliers_iso)`
	`print(f"Number of outliers (Isolation Forest): {num_outliers_iso}")`
	
	`# PCA plot with marked outliers (Isolation Forest)`
	`plt.figure(figsize=(20, 10))`
	`plt.scatter(X_train_pca[:, 0], X_train_pca[:, 1], label='Data points', alpha=0.6)`
	`plt.scatter(X_train_pca[outliers_iso, 0], X_train_pca[outliers_iso, 1], color='r', label='Outliers', edgecolors='k')`
	
	`plt.xlabel('Principal Component 1')`
	`plt.ylabel('Principal Component 2')`
	`plt.title('PCA Plot with Outliers Marked (Isolation Forest)')`
	`plt.legend()`
	`plt.grid()`
	`plt.show()`