ABSTRACT

Skin diseases are very common for humans in day-to-day lives. The task of recognizing the type of skin disease and prediction of its occurrence is the most complicated one as we have hundreds of varieties of skin textures and different visual closeness effects of diseases and other factors. Therefore, it is of high importance to detect the occurrence of any skin disease and also, we should be able to find the type of skin disease to get best treatment.

The proposed system focuses on building the model for the prediction and classification analysis of different types of skin diseases using deep learning techniques and algorithms. These deep learning algorithms help the model in predicting and classifying the skin diseases on different characteristics of the dataset with more accuracy.

CONTENTS

List of Abbreviations

AI – Artificial Intelligence

ML – Machine Learning

NN – Neural Networks

ANN – Artificial Neural Networks

CNN – Convolutional Neural Networks

SVM – Support Vector Machine

KNN – K-nearest neighbors

RGB – Red Green Blue

EDA – Exploratory Data Analysis

IDA - Integrated Data Analysis

CSV – Comma Separated Values

DML – Data Manipulation Language

SIFT – Scale Invariant Feature Transform

OVR – One Vs Rest

OVO – One Vs One

PCA – Principle Component Analysis

AUC -Area under the Curve

ROC – Receiver Operating Characteristics curve

TP – True positive

TN – True Negative

FP – False Positive

FN – False Negative

TPR – True Positive Rate

FPR – False Positive Rate

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Dermatological problems are the major widespread diseases in the world that are very common in our daily lives. The diagnosis of these skin diseases is highly complicated and is not accurate due to various kinds and complexities in skin complexions, tone, and presence of hair on the skin, etc., So we identify dermatology as one of the most complicated branches of science [1]. We have these variations and varieties in skin diseases and skin types due to many changes made in the environment, geographical changes, biological changes, pollution, etc., These skin diseases can spread from one person to another. So, it is important to treat them at the early stages to avoid problems. The total well-being of a person both physically and mentally is also affected due to these skin diseases in a wide range. There might be a threat to life sometimes due to the severity of the disease. The prevalence of skin diseases has increased over the past few decades and they contribute to a significant burden on healthcare systems across the world. So, there is a need to have an instrument or model to detect and identify the type of skin disease at the earliest stage possible and give proper treatment for each type of skin disease without facing unexpected life risk issues.

According to various studies and surveys, the existence of skin diseases among the general population varied from 8% to 11%. In the year 2017, the number of age-standardized years in India who lived with disability for cardio-vascular diseases was nearly 333 and for skin and subcutaneous diseases was nearly 455 per 100,000. As per the global burden of disease study in 2017, cardiovascular diseases ranked 12th whereas skin diseases ranked 10th based on these age-standardized years lived with disability.

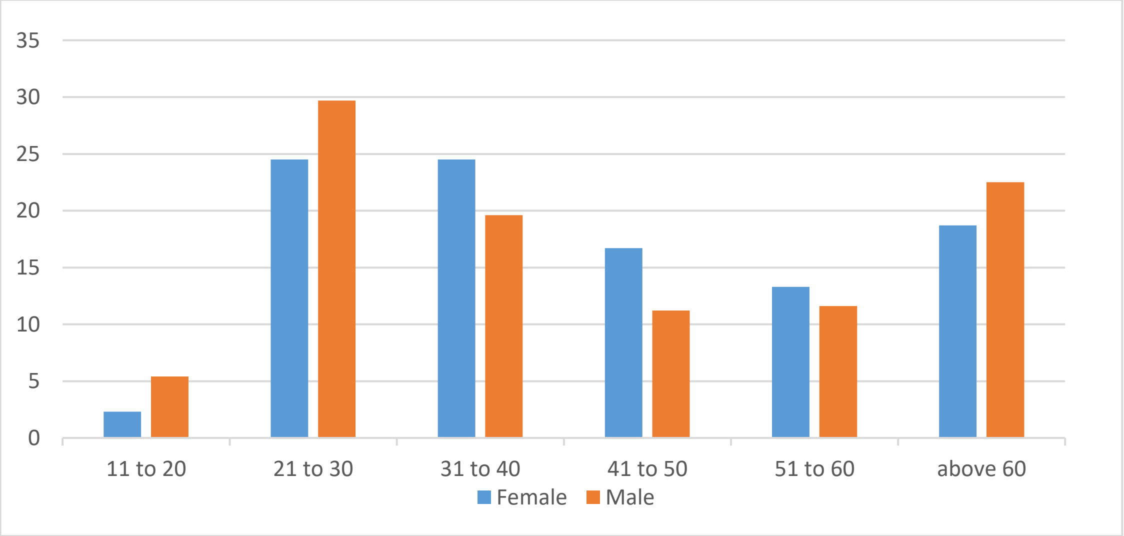


Figure 1.1: Bar diagram showing distribution of sex according to the age

Table 1.1: Sex distribution according to the age of the surveyed population

Age in years	Female	Male	Total
11-20	8(2.3)	15(5.4)	23(3.7)
21-30	85(24.5)	82(29.7)	167(26.8)
31-40	85(24.5)	54(19.6)	139(22.3)
41-50	58(16.7)	31(11.2)	89(14.3)
51-60	46(13.3)	32(11.6)	78(12.5)
60+	65(18.7)	62(22.5)	127(20.4)
Total	347(55.7)	276(44.3)	623(100.0)

Table 1.1 and **Figure 1.1** shows the survey results of people based on their age and sex for ages ranging from 11 and above. From the results of the survey, we can say that there are 55.7% females and 44.3% males. Almost half of the people i.e., 52.8% are of the age less than 41 years and the remaining people of age 60 years and above constitute 20.8%.

1.1 Identification of seriousness of the problem

The diagnosis of these skin diseases is highly complicated due to various kinds and complexities in skin complexions, tone, and presence of hair on the skin, etc., These skin diseases are infectious and need to be treated at earlier stages to avoid spreading from one person to other. The total well-being of a person both physically and mentally is also affected due to these skin diseases in a wide range. There might be a threat to life sometimes due to the severity of the disease. So, there is a need to have an instrument or model to detect and identify the type of skin disease at the earliest stage possible and give proper treatment for each type of skin disease without facing unexpected life risk issues.

According to various studies and surveys, the existence of skin diseases among the general population varied from 8% to 11%. In the year 2017, the number of age-standardized years in India who lived with disability for cardio-vascular diseases was nearly 333 and for skin and subcutaneous diseases was nearly 455 per 100,000. As per the global burden of disease study in 2017, cardiovascular diseases ranked 12th whereas skin diseases ranked 10th based on these age-standardized years lived with disability.

1.2 Problem definition

The data is collected from various sources and surveys to find patterns in the attributes and factors causing skin disease. The important factors like age, family history, itching factor, and some other factors are focused mainly in the data. The primary concern is to analyze these important factors in order to predict the occurrence of a skin disease and the type of skin disease occurred.

1.3 Objective

With the help of the model, we construct for identifying the skin diseases in humans, we intend to find out the patterns in the data gathered by analyzing different factors which helps to achieve our goal of prediction and classification of the skin disease that occurred so that we can treat it at the earliest possible stage and avoid major life risks.

1.4 Existing models

There are many researchers and models previously and some are still going on to identify skin diseases in different regions of a human being. But it doesn’t focus on certain issues to be analyzed like family history, intensity of disease etc., and also prediction and classification with good accuracy is also not so good.

CHAPTER 2
LITERATURE SURVEY

Skin diseases are hazardous to life. They should be detected at an early stage to cure them properly without spreading. But the task of detection is very complicated. Several researchers proposed various methodologies and technologies for accurate and appropriate detection of skin diseases.

In [2], some researchers suggested skin disease detection on the basis of image processing. Here they used Red Green Blue spectrum (RGB) images of the areas affected by skin diseases as input. These images are re-sized and features are extracted using a pre-trained CNN (Convolution Neural Network) model and then they applied multi support vector machine (SVM) to classify. Finally, they have shown that this is the simplest method with 100% accuracy. In [3], researchers have suggested a model using SVM and KNN algorithms for the purpose of segmentation and classification of a skin disease occurred. In other paper [4], Researchers used an artificial intelligence (AI) system on the basis of neural network (NN) concepts. This system is divided into two parts. They are feature extraction and classification. Feature extraction is done by image acquisition and classification is done by feed-forward neural network.

Other researchers in paper [5] studied six different algorithms for the classification of different skin diseases. They chose 15 most important features for prediction of the disease class to which it belongs. In addition to these 6 algorithms, some authors also created an ensemble method using the combination of other classifier techniques too. At last, keeping all the observations in mind, they concluded that more accurate and very effective skin disease detection and prediction is done using the ensemble methods.

In [6] a researcher used three techniques – Segmentation which is done by thresholding, Extraction of features using two-dimensional (2-D) wavelet decomposition and then classification performed using radial basic neural network and back propagation neural network to achieve the results. Researchers in [7] have proposed a model on the basis of adaptive federated machine learning technique for skin diseases detection. This process has an architecture of intelligent local edges and a central (global point) known as a server to diagnosis the type of skin, skin diseases and also to improve the accuracy.

In [8] researchers have introduced a model using the methods of extracting features and classification of images for detection of six different types of skin diseases and their conditions.

In [9], other researcher has proposed a system using ad boost classifier and statistical analysis for identification of skin problem at an early stage. Their research focused mainly on identifying the symptoms of skin cancer at an early stage on the basis of statistical analysis obtained by using correlation algorithms. In [10], researcher has proposed a model on the basis of feature extraction of skin image using color texture and to identify the skin disease they used segmentation and SVM. In [11], researcher has proposed a system based on computer vision technique for skin disease detection which could be implemented on both mobile and computer using desktop applications.

In another research paper [12], the researchers used five machine learning algorithms Logistic Regression, random forest, Naïve Bayes, kernel SVM, and Convolution Neural Network algorithms for skin diseases detection. Finally, from the confusion matrix, they found that the convolution neural network (CNN) model is the best model for skin disease detection. In [13], some other researchers have proposed the usage of various kinds of image processing algorithms for feature extraction and feed forwarding using artificial neural network to train and test the model. This work is of two parts, feature extraction and classification. The feature extraction takes place on the basis of color texture and the classifier finds the possible skin disease. In [14] researchers have proposed a model to identify the psoriasis using color feature extraction and classification of the skin disease. In this paper, Abbadi has mentioned color feature extraction method where color features are extracted using a mathematical formula used for RGB color value and texture extraction method where texture features are extracted using various components such as contract, energy, entropy, and homogeneity of the skin image. After that Neural Network (NN) algorithms are used to detect the psoriasis present on the skin. In [15] researchers have proposed a model using machine learning and computer vision concepts. The feature extraction of images is done and various algorithms are applied on these features for the detection of six types of skin diseases with an accuracy of 95%. In [16] a researcher has suggested an image clustering method using naïve bayes classifier. They have made use of Scale Invariant Feature Transform (SIFT) method to detect key points of image. Later they used SVM and CNN for the tasks of segmentation and classification with 84% accuracy and 82% precision. In [17], the authors proposed an artificial intelligence (AI) system consisting of image processing and a deep neural network for the detection of skin cancer. First, they performed segmentation on the affected area and extracted features using image processing, and used a convolution neural network for prediction. This resulted in 93.7% accuracy for training dataset and 89.5% accuracy for testing dataset. In [18], researchers have proposed a two-stage method for skin disease prediction on the basis of color texture-based identification and used a classification technique to identify the disease name. The first stage has 95.99% accuracy and the second stage has a little less accuracy compared to the first stage i.e., 94.016%. In [19], a researcher proposed that features are extracted with the help of graph cut algorithms which are used for image processing of skin images. For classification, he used naïve Bayes algorithm as a classification algorithm.

CHAPTER 3
THEORETICAL BACKGROUND

3.1 Machine learning Vs Deep learning

3.1.1 What is Machine Learning?

Machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it to learn for themselves. Machine learning is something that is capable to imitate the intelligence of the human behavior. Machine learning is used to perform complex tasks in a way that humans solve the problems. Machine learning can be descriptive it uses the data to explain, predictive, and prescription.

3.1.2 Why Machine Learning?

Machine learning involves computers learning from data provided so that they carry out certain tasks. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step. The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithms it uses to determine correct answers.

The nearly limitless quantity of available data, affordable data storage, and growth of less expensive and more powerful processing has propelled the growth of ML. Now many industries are developing more robust models capable of analysing bigger and more complex data while delivering faster, more accurate results on vast scales. ML tools enable organizations to more quickly identify profitable opportunities *and* potential risks.

The practical applications of machine learning drive business results which can dramatically affect a company’s bottom line. New techniques in the field are evolving rapidly and expanded the application of ML to nearly limitless possibilities. Industries that depend on vast quantities of data—and need a system to analyse it efficiently and accurately, have embraced ML as the best way to build models, strategize, and plan.

3.1.3 What is Deep Learning?

Deep learning is the subset of the machine learning technique. It is also known as deep neural network because it uses the architecture of neural network. Deep learning eliminates some of data preprocessing. This has one input layer and one output layer and more than one hidden layer. This uses labeled data for training deep learning prediction is more accurate. Deep learning is an element of data science, it includes statistics and predictive modeling. This approach is beneficial to the person whose task are related to collecting, analyzing large amounts of data.

3.1.4 Why Deep Learning?

Algorithms used in deep learning learn high level features for data. The more the data you fed to the deep learning the better the result will be. Deep learning handles large volumes of data and it is kept to the best use when it comes to the large sets of unstructured data. Deep learning has less accuracy when it is fed with the less data.

3.2 Machine Learning Approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

3.2.1 Supervised learning

Supervised learning is one of the machine learning approaches through which models are trained using perfectly labelled training data and on the basis of that models predicts the output. Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are.

3.2.2 Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labelled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

3.2.3 Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that un-labelled data, when used in conjunction with a small amount of labelled data, can produce a considerable improvement in learning accuracy. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

3.2.4 Reinforcement learning

Reinforcement learning is feedback-based machine learning technique, and it aims to maximize the rewards by their hit and trial actions. In reinforcement learning the model learns automatically using feedbacks without any labeled data, unlike supervised learning and since there is no labelled data, the model is used to learn from its experiences only.

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

3.3 Machine Learning Models

Performing machine learning involves creating a model, which is trained on some training data and then can process additional data to make predictions. Various types of models have been used and researched for machine learning systems.

3.3.1 Artificial neural networks

Artificial neural networks (ANN) is a sub field of artificial intelligence which is simply known as neural networks. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes. ANN consists of 3 layers they are input layer, output layer and hidden layer. Input layer accepts inputs in several different formats provided by the programmer. The hidden layer presents in between input and output layers. It performs all the operations to find hidden features and patterns. Output layer provides the output based on all the calculations provided by the hidden layer.

3.3.2 Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

3.3.2.1 Linear SVM

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and the classifier is used as Linear SVM classifier.

3.3.2.2 Non-linear SVM

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and the classifier used is called Non-linear SVM classifier. In its most simple type, SVM doesn’t support multiclass classification natively. It supports binary classification and separating data points into two classes. For multiclass classification, the same principle is utilized after breaking down the multi classification problem into multiple binary classification problems. The idea is to map data points to high dimensional space to gain mutual linear separation between every two classes. This is called a One-to-One approach, which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes. Another approach one can use is One-to-Rest. In that approach, the breakdown is set to a binary classifier per each class.

3.3.3 Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems.

3.3.3.1 Types of Logistic Regression

Based on the number of categories of target variables that can be predicted, Logistic regression can be divided into following types –

- **Binary or Binomial** In such a kind of classification, a dependent variable will have only two possible types either 1 and 0.

- **Multinomial** In such a kind of classification, dependent variables can have 3 or more possible unordered types or the types having no quantitative significance.
- **Ordinal** In such a kind of classification, dependent variables can have 3 or more possible ordered types or the types having a quantitative significance.

3.3.4 k-Nearest Neighbors Algorithm

The k-nearest neighbors' algorithm (k-NN) is a non-parametric classification method and is used for classification and regression. In both cases, the input consists of the k closest training examples in the data set. The output depends on whether k-NN is used for classification or regression: In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors' (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearest neighbors contribute more to the average than the more distant ones.

3.3.5 Naïve Bayes Classifier

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of colour, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of **Bayes' Theorem**

Bayes' Theorem:

Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

- The formula for Bayes' theorem is given as:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

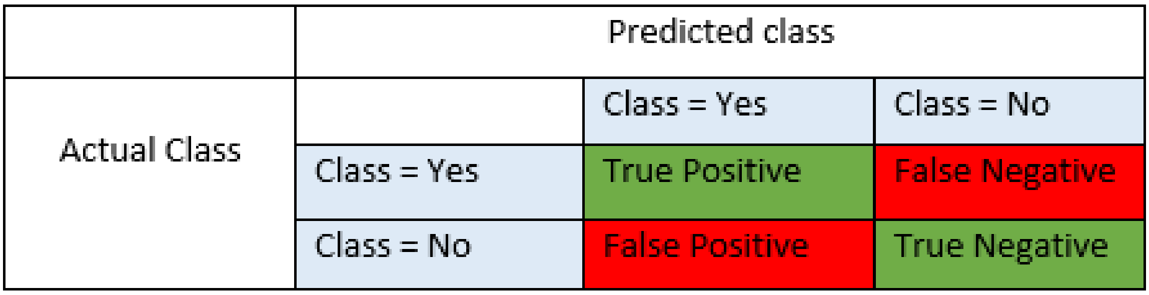
- **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc., The classifier uses the frequency of words for the predictors.
- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word

is present or not in a document. This model is also famous for document classification tasks.

3.4 Confusion matrix and its metrics

3.4.1 Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. All the measures can be calculated by using left most four parameters. So, let’s talk about those four parameters first.



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 3.1: Confusion matrix with 4 parameters

True positive and true negatives are the observations that are correctly predicted. We want to minimize false positives and false negatives. These terms are a bit confusing. So, let’s take each term one by one and understand it fully.

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP) – When actual class is no and predicted class is yes.

False Negatives (FN) – When actual class is yes but predicted class in no.

3.4.2 AUC- ROC curve

The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR** against **FPR** at various threshold values and essentially **separates the 'signal' from the 'noise'**. The **Area Under the Curve (AUC)** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives. When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points.

3.4.3 Metrics

- **Accuracy**

  Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

  Accuracy = TP+TN/TP+FP+FN+TN

- **Precision**

  Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

  Precision = TP/TP+FP

- **Re-call/ Sensitivity/ True Positive Rate (TPR)**

  Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

  Recall = TP/TP+FN

- **F1 Score**

  F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

- **False Positive Rate (FPR)**

  FPR tells us what proportion of the negative class got incorrectly classified by the classifier.

  FPR=FP/TN+FP

3.5 Exploratory Data Analysis

Exploratory data analysis is an approach of analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. EDA is different from initial data analysis (IDA) which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA. Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.

CHAPTER 4
APPROACH DESCRIPTION

4.1. Approach Flow

1. We collected the data related to the dermatology by researching through the online website [20].
2. Import the required packages for the classification model.
3. Creating a project and loading the dataset from the comma separated values (CSV) file which is based on analysis of different variables.
4. Data cleaning:
    - Check if there are any duplicate rows.
    - Check the presence of any null values in each column.
    - Overview the dermatology dataset (optional)
    - Rename the 35 columns with the following names in sequential order: 'erythema','scaling','definite_borders','itching','koebner_phenomenon', 'polygonal_paples','follicular_papules','oral_mucosal_involvement','knee_elbow_involvement','scalp_involvement','family_history', 'melanin', 'eosinophils', 'PNL', 'fibrosis', 'exocytosis', 'acanthosis', 'hyperkeratosis', 'parakeratosis', 'clubbing', 'elongation', 'suprapapillary_epidermis', 'spongiform_pustule', 'munro_microabcess', 'focal_hypergranulosis',' granular_layer', 'basal_layer', 'spongiosis’,’sawtooth', 'follicular', 'perifollicular', 'inflammatory_monoluclear', 'band-like infiltrate', 'Age', 'target'
5. EDA - Exploratory data analysis (EDA) is a method to analyze the data sets and acquire the summary of important attributes and their relationships through various forms and techniques of data visualization based on some statistical data calculated.

    a. Count of Different Target Values (Skin Disease Vs Count)

    b. Distribution of Age According to Disease

    c. Is the Disease Due to Family Genes? (Family History Vs Count)

    - Family History=Yes Vs Count
    - Family History=No Vs Count
6. Correlation heatmap of complete dataset.

7. Machine Learning – Classification (Define and design optimal model by using Naïve Bayes classification model)

   a. Building Feature set

   b. Naïve Bayes model

8. Training the model using the pre-trained models which were trained and tested using the dataset.

9. Testing the model

10. Checking result and Drawing inferences

CHAPTER 5

DATA EXPLORATION

5.1. Dermatology dataset

The data we handle is Dermatology dataset. This dataset consists of 366 persons data of the skin diseases categorized into 6 different classes. The dataset consists of 35 columns in total and here we have 34 columns which are attributes based on which we can determine the “target” column which represents the type of skin disease occurred. Some of the important attributes are “age”, “family_history” and “target” (which has numbers ranging from 1 to 6 each representing different type of skin disease).

The entire dataset is being operated to train the model. Later this dataset is split into training set and testing datasets. Testing set will calibrate the parameters and is used only to get the performance and efficiency of the system.

5.1.1. Data Manipulation

Manipulation of data is the process of manipulating or changing the information to make it more structured and understandable to work easily on the data. We use Data manipulation language (DML) to accomplish this task.

Here are the steps that are to be considered to get commenced with data manipulation:

1. Data manipulation is only possible when you have data to do so. Therefore, you need a database which is generated from various data sources.
2. Manipulation of data helps in cleaning the information. This work requires the data in database to be re-organized and re-structured.
3. First of all, we need to import the database to perform the work on the data.
4. We can also insert, remove, and make changes in the databases and its information with the help of data manipulation step.
5. One of the important steps of our project i.e., data analysis becomes simple after performing the data manipulation step.

The information or data is in different formats (like .csv, .txt, .xlsx) so it is converted into a standard format that is all the data is converted in .csv format.

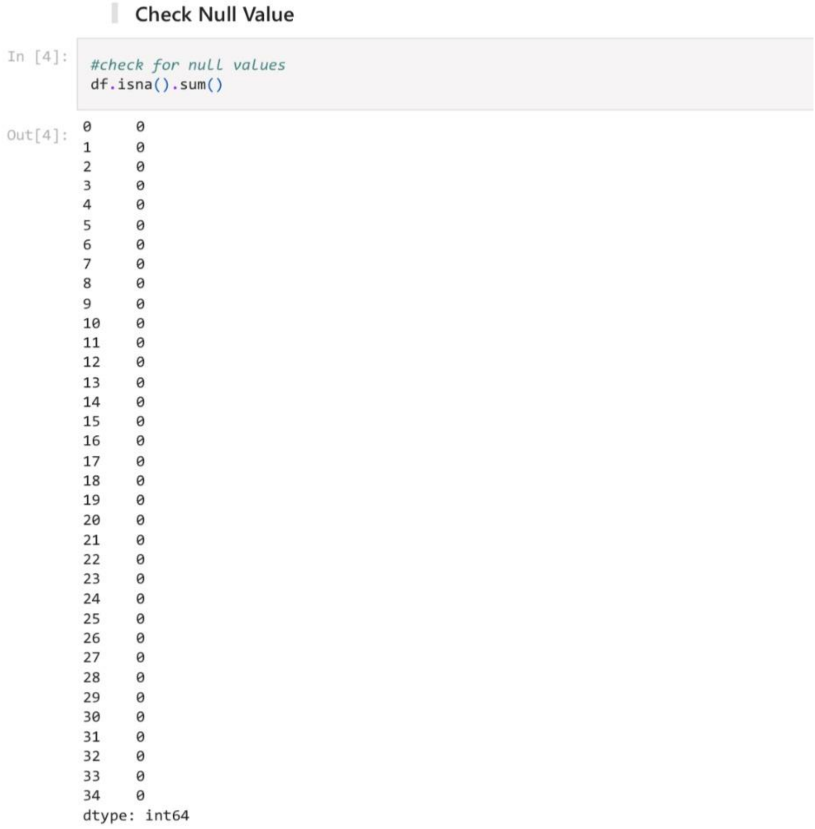
5.1.2. Data Preparation

Data preparation is the one of the important steps to be done before we analyse the data. It is the process of cleaning the raw data available in the dataset before processing and analysis. It also involves reformatting, correcting mistakes and the consolidating the data sets to polish data.

5.1.2.1. Missing Values

We should check whether there are missing values in any of the columns and should handle them properly.

Generally missing values are either filled with random values leading to less accuracy or they are ignored which is not good if half of the data contains missing values. The efficient way to fill these missing values is by finding the correlation between the other attributes.



Check Null Value

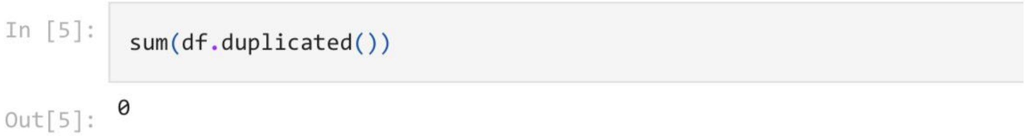
```
In [4]:  #check for null values
         df.isna().sum()

Out[4]:  0     0
         1     0
         2     0
         3     0
         4     0
         5     0
         6     0
         7     0
         8     0
         9     0
         10    0
         11    0
         12    0
         13    0
         14    0
         15    0
         16    0
         17    0
         18    0
         19    0
         20    0
         21    0
         22    0
         23    0
         24    0
         25    0
         26    0
         27    0
         28    0
         29    0
         30    0
         31    0
         32    0
         33    0
         34    0
         dtype: int64
```

Figure 5.1: Checking for number of missing values in each column

5.1.2.2. Duplicate Values

We should check whether there are duplicate or repeated values in any of the rows and should handle them properly.

Generally, it is good to remove the duplicate rows which increases accuracy and saves time.

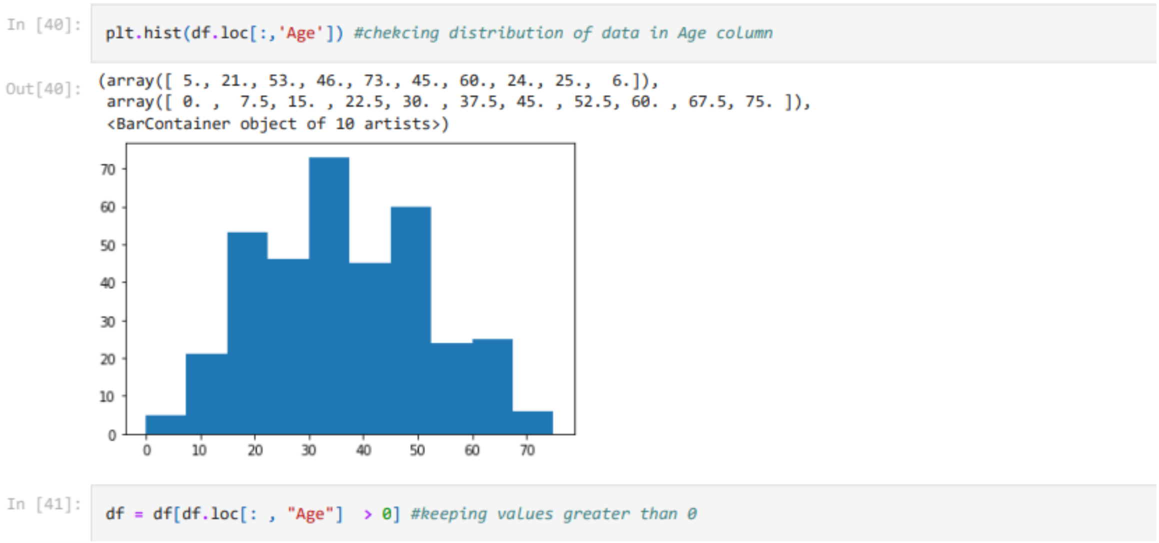


```
In [5]:  sum(df.duplicated())

Out[5]:  0
```

Figure 5.2: Checking for number of duplicate rows in dataset

5.1.2.3. Outliers

Outliers are the abnormal data points which don't belong to a particular data population. Their values lie far away from other values. An outlier simply depicts the mistakes made in measurements. An outlier is an observation that differ from the well-structured and well-organized data. We use boxplots, histograms and many other visualisation techniques to find outliers in the **numeric data**. In the case of **categorical data**, it is highly impossible for outlier detection.

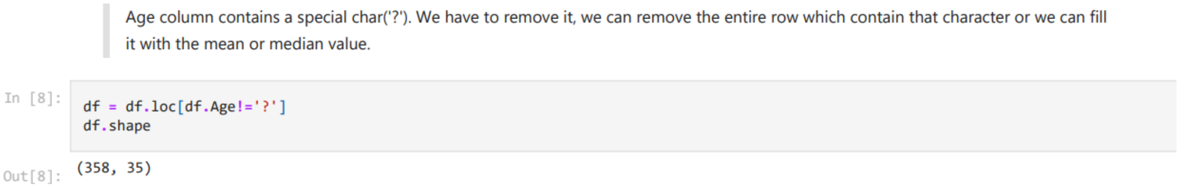


```
In [40]: plt.hist(df.loc[:,'Age']) #chekcing distribution of data in Age column

Out[40]: (array([ 5., 21., 53., 46., 73., 45., 60., 24., 25.,  6.]),
          array([ 0. ,  7.5, 15. , 22.5, 30. , 37.5, 45. , 52.5, 60. , 67.5, 75. ]),
          <BarContainer object of 10 artists>)
```

```
In [41]: df = df[df.loc[: , "Age"]  > 0] #keeping values greater than 0
```

Figure 5.3: Check and remove outliers in age column with histogram

Here we are removing an outlier in age column by checking if there is any person of age<0 which is impossible and is taken as an abnormal case.



> Age column contains a special char('?'). We have to remove it, we can remove the entire row which contain that character or we can fill it with the mean or median value.

```
In [8]: df = df.loc[df.Age!='?']
        df.shape

Out[8]: (358, 35)
```

Figure 5.4: Check and remove outliers like special chars in age column

Here we are removing an outlier in age column by checking if there is any person of age containing special character "?" which is impossible for analysis of skin disease and is taken as an abnormal case.

CHAPTER 6

DATA ANALYSIS

6.1. Exploratory data analysis (EDA) and its types

Exploratory data analysis (EDA) is a method to analyze the data sets and acquire the summary of important attributes and their relationships through various forms and techniques of data visualization based on some statistical data calculated.

These summaries are of 2 types:

1. **Numerical:** These summaries will be in the form of numbers.

   Ex: Average (Mean), Mode, Median, Summation etc.,

   Depending on the number of variables, numerical summary is of three types:

   - Univariate
   - Bivariate
   - Multivariate

2. **Graphical:** Here the summaries are represented by graphs.

   Ex: Counter plot, bar graph, histogram, etc.,

6.2. Why do we need Data Analysis?

It is important to analyse the data for the below reasons:

1. To identify the distribution of dataset.
2. Selecting the appropriate machine learning model and algorithm.
3. Best features extraction.
4. Evaluation of the model and presentation of the test results.

6.3. EDA Inferences

6.3.1. Count of Different target values (Skin Disease Vs Count)

As we mentioned earlier, there are 6 different types of values in “target” each representing 6 different types of diseases. We plot a counter plot showing the count of number of persons having each type of skin disease.

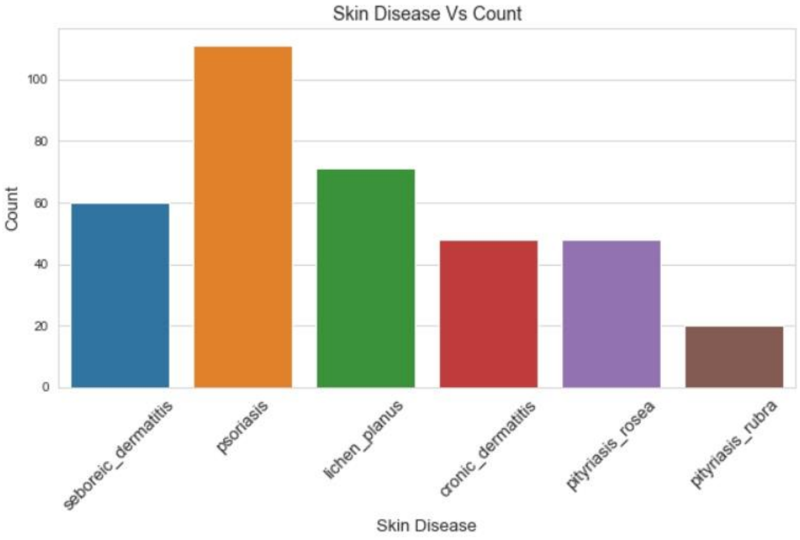


Figure 6.1: Counterplot for Skin disease Vs Count

6.3.2. Distribution of Age according to Disease

As we mentioned earlier, there are 6 different types of values in “target” each representing 6 different types of diseases. We plot a Kernel Density Estimate (KDE) plot showing the distribution of age of persons having each type of skin disease.

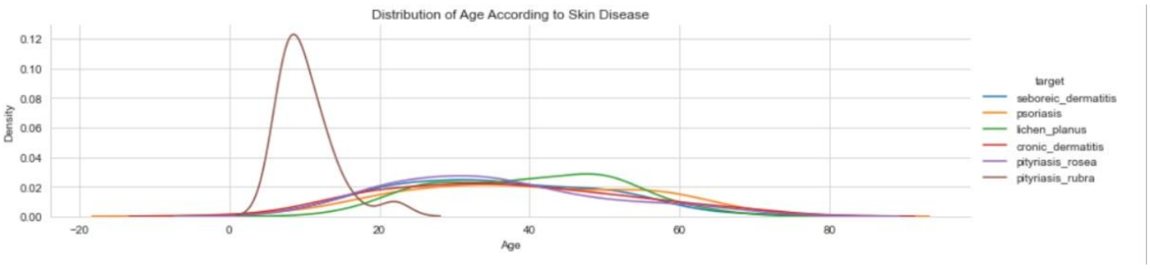


Figure 6.2: Distribution of Age according to the Skin disease

6.3.3. Itching factor Vs Skin disease

As we mentioned earlier, there are 6 different types of values in “target” each representing 6 different types of diseases. We plot a Kernel Density Estimate (KDE) plot showing the distribution of itching factor of persons having each type of skin disease.

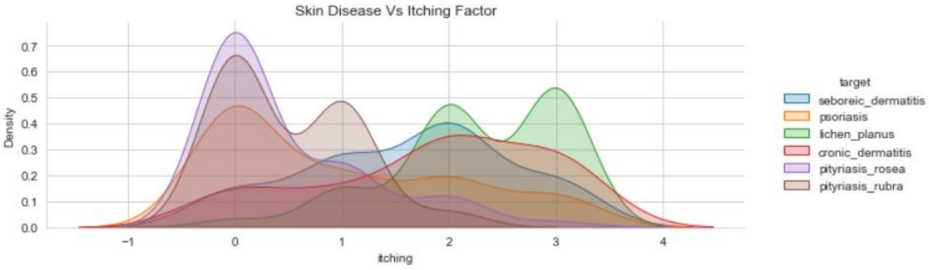


Figure 6.3: Itching factor Vs Skin disease

6.3.4. Family history Vs Count

We plot a counter plot showing the number of persons having skin disease due to family history and number of people having skin disease not because of family history.

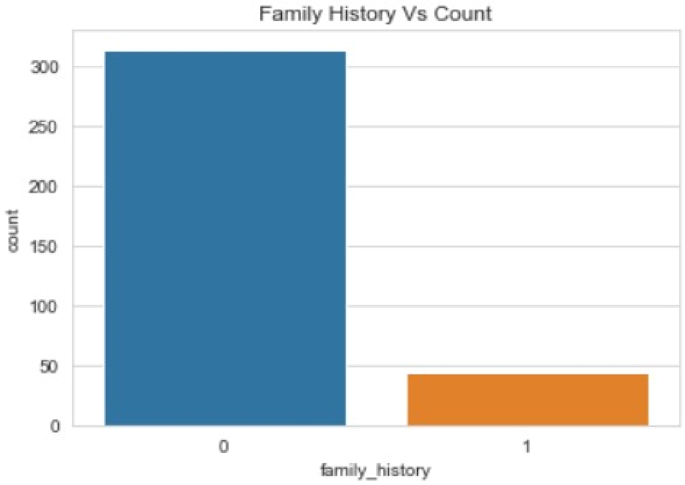


Figure 6.4: Family history Vs Count

6.3.4.1. Family history-YES Vs Count of each disease

We plot a counter plot showing the number of persons having different types of skin diseases due to family history.

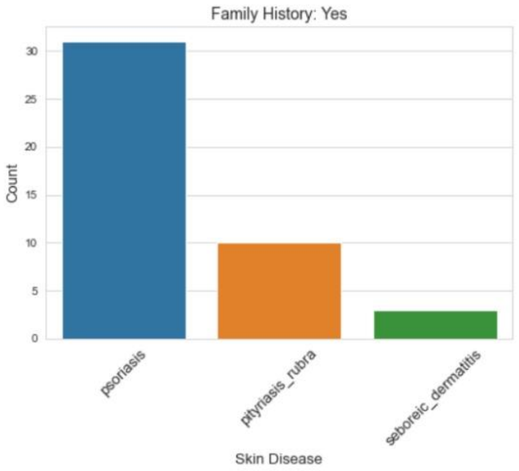


Figure 6.5: Family history-yes Vs Count

6.3.4.2. Family history-NO Vs Count of each disease

We plot a counter plot showing the number of persons having different types of skin diseases but are not due to family history.

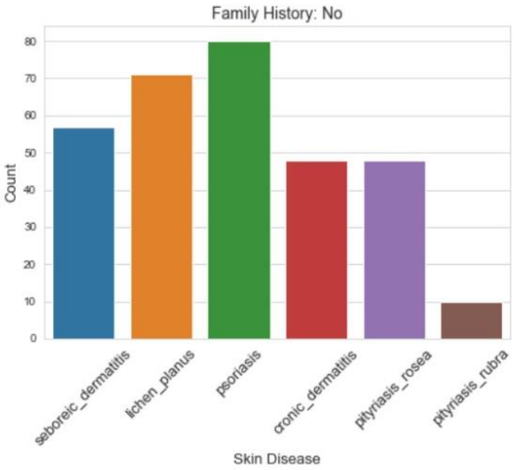


Figure 6.6: Family history-no Vs Count

6.4. Correlation Heatmap

Correlation heatmaps are the plots that are used to show the visualization of the strength of the relationship between numerical variables. These plots are also used to estimate which variables are related to each other and how strong is their relationship. A correlation plot can find both linear and nonlinear relationships.

Correlation plots contain rows and columns. The columns contain the names of numerical variables and the rows depicts the relationships between all pairs of variables. Each value in each cell represents the strength of the relationships.

These relationships are of two types:

1. Positive relationship – If the values are positive
2. Negative relationship – If the values are negative

Below is a diagram showing the correlation heatmap to understand the relationships and their strength between each variable to all other variables in the dataset by using color-coding technique. The color-coding technique helps to easily identify the variables and their relationships just by a glance. In the color scale dark green represents strong relation between variables and dark red represents weak relations.

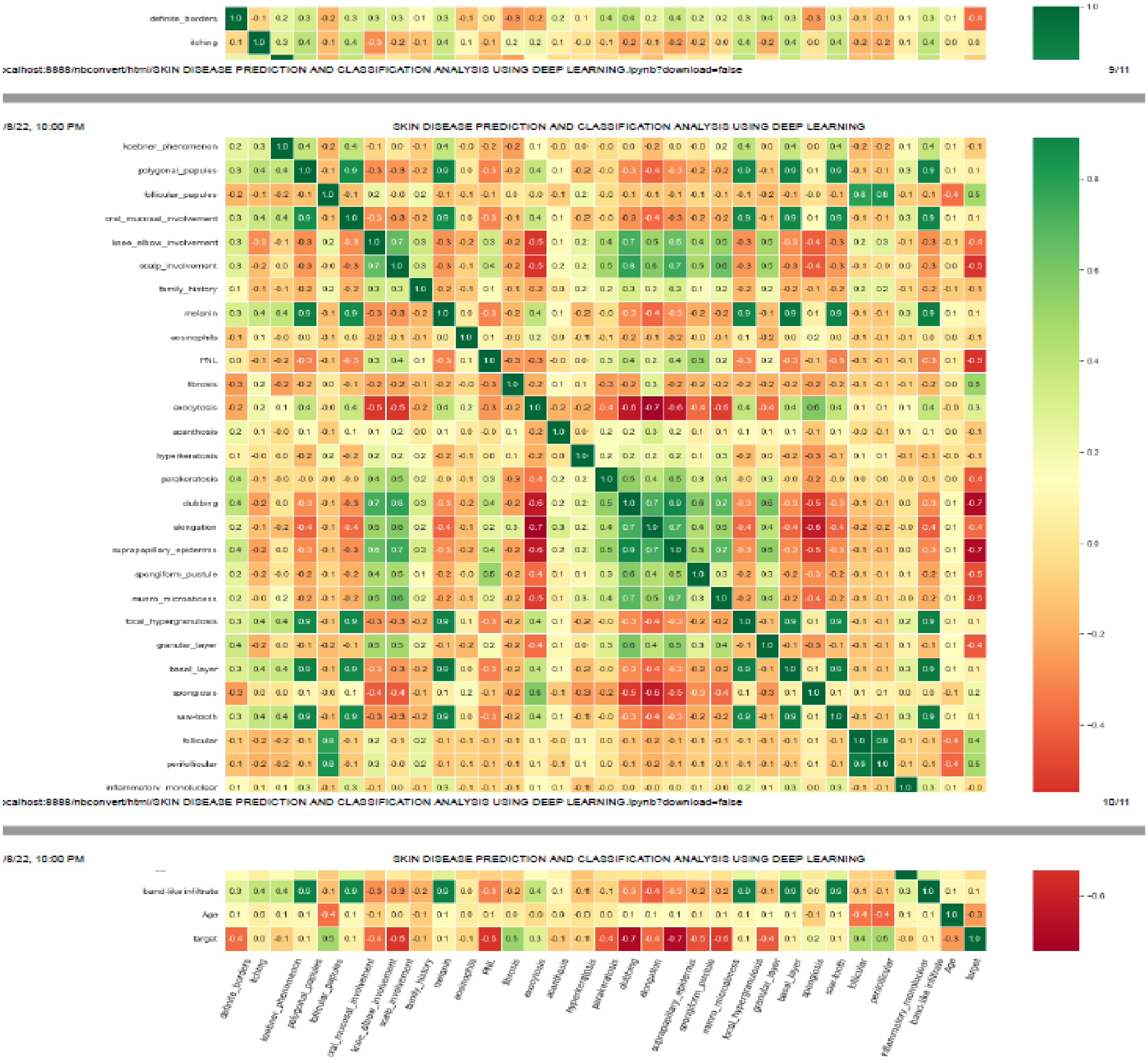


Figure 6.7: Correlation Heatmap

CHAPTER 7
MODELLING

7.1. Model Development

7.1.1. Multi-class classification problem

Multiclass or **multinomial classification** is a problem which classifies the instances into one of the three or more classes (classification of instances into one of the two classes is called binary classification).

While there are many classification algorithms like multinomial logistic regression naturally permit the use of more than two classes, some are by nature binary algorithms. These algorithms can, however, be turned into multinomial classifiers by several number of strategies.

7.1.2. Transformation to binary

To reduce the problem of multiclass classification to multiple binary classification problems there are some strategies. It can be categorized either into one vs rest or one vs one. These techniques are developed on the basis of reduction of the multi-class problem into multiple binary problems. These techniques are also called as problem transformation techniques.

One-vs.-rest

One-vs.-rest (OVR) is a strategy of training a single classifier per class, with the samples of that class as positive and negative samples. Rather than just a class label, this requires the base classifiers to produce a real-valued confidence score for making decisions; discrete class labels alone can cause inconsistencies, whereas we can predict multiple classes for a single sample.

Here is a pseudocode of the training algorithm for an OVR learner built from a binary classification learner L:

```
Inputs:

    • L is a learner (training algorithm for binary classifiers)
    • X is samples
    • y labels where $y_i \in \{1, \dots K\}$ represents the label for the sample $X_i$
Output:

    • $f_k$ is a list of classifiers where $k \in \{1, \dots, K\}$
```

Procedure:

- For each value of k in {1, …, K}
    - We need to construct a new label vector z where,

      If $y_i = k$, then $z_i = y_i$ else it is 0

    - Apply L to X, z so as to obtain the list of classifiers f_k

Corresponding classifier reports the highest confidence score for which making decisions means applying all classifiers to an unseen sample x and predicting the label k.

One-vs.-one

One-vs.-one (OVO) reduction is a strategy of training K(K-1)/2 binary classifiers to reduce the problem of K-way multiclass classification. From the original training set, each classifier receives the pair of class samples and also must learn to differentiate between these two classes. At the time of prediction, we prefer voting. In this voting, these K(K-1)/2 classifiers are assigned to an unknown and unseen sample and finally the class which got the highest number of "+1" predictions is predicted by the combined classifier.

7.1.3. Extension from binary

To address the problem of multi-class classification, many algorithms have been evolved on the basis of decision trees, naïve bayes, k-nearest neighbors, support vector machines, neural networks and extreme machines for learning. We use Naïve Bayes for the creation of our model.

7.1.3.1. Naive Bayes

Naive Bayes is one of the successful classifiers which is based on MAP (Maximum a posterior) estimation.

The Naïve Bayes algorithm contains two words Naïve and Bayes, which are described as below:

Naïve: It is referred as Naïve, because it assumes in the way for certain features occurrence is independent of other features occurrences. Such as if the fruit is recognized on the bases of colour if it red, and taste if it is sweet and shape if it is spherical, then that fruit is recognized as an apple. Hence every feature contributes individually it's part to identify whether it is an apple or not, without relying on other features.

Bayes: As it depends on the law of Bayes' theorem, it is known as Bayes.

Bayes' Theorem:

Bayes' theorem is also known as Bayes' rule or Bayes' law. It depends on the conditional probability. Bayes' theorem is helpful to determine the probability of a hypothesis with earlier knowledge.

- The formula for Bayes' theorem is:

$$\mathrm{P(A|B)}= \frac{P(B|A)P(A)}{P(B)}$$

Here,

P(A|B): Posterior probability which means it is the probability of hypothesis A on the observed event B.

P(B|A): Likelihood probability which means it is the probability of the evidence given that the probability of a hypothesis is true.

P(A): Prior Probability which means it is the probability of hypothesis before observing the evidence.

P(B): Marginal Probability which means it is the probability of Evidence.

7.2. Model Evaluation

We have created the model and trained it using Naïve Bayes classifier. Now, our task is to test the testing dataset and predict different metrics like precision score, accuracy, re-call, f1 score and ROC-Curve to determine how well our model is being worked.

CHAPTER 8
RESULTS AND CONCLUSIONS

8.1. Results

1. We can see that age is the most important factor affecting the skin disease. In our project we can determine that the persons between age group 10-30 are having high chance of getting pityriasis_rubra.
2. There are nearly 40 people who suffered from skin disease due to family history and more than 300 people who suffered from disease but not due to family history.
3. The people having a background of family history are most likely to suffer from the diseases like pityriasis_rubra, seoboriec_dermatitis, psoriasis.
4. Based on the count of different target values, we can conclude that most of the people suffered from the disease named “psoriasis”.
5. The precision, re-call and f1 scores obtained for the 6 classes in our project are shown in the below table:

Table 8.1: Results of Performance metrics of each class label

	Precision	**Recall**	**F1 Score**
Class 0	0.971	0.971	0.971
Class 1	1.0	0.864	0.927
Class 2	1.0	1.0	1.0
Class 3	0.526	0.909	0.667
Class 4	1.0	0.923	0.96
Class 5	1.0	1.0	1.0

8.2. Conclusion

In our project, we have predicted and classified skin diseases using deep learning algorithms. It is found that we can combine various features and deep learning algorithms to achieve high accuracy rate and we can also combine various models done previously for the prediction of many other diseases. The previous models were able to report six skin diseases with maximum accuracy of 75%. With the usage of deep learning algorithms, we can predict skin diseases with more accuracy i.e.,99.31%. This shows that deep learning algorithms have a great potential in the real-world diagnosis of skin disease. This work can be extended in future by automating the skin disease diagnosis process as it will save time and cost.

CHAPTER 9

REFERENCES

[1] Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra and B. Hu, "Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends," 2015 IEEE 8th International Conference on Cloud Computing, 2015, pp. 621-628, doi: 10.1109/CLOUD.2015.88. https://ieeexplore.ieee.org/document/7214098

[2] Nawal Soliman ALKolifi ALEnezi, A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science, Volume 163, 2019, Pages 85-92, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.12.090 (A Method Of Skin Disease Detection Using Image Processing And Machine Learning - ScienceDirect)

[3] R. Sumithra, Mahamad Suhil, D.S. Guru, Segmentation and Classification of Skin Lesions for Disease Diagnosis, Procedia Computer Science, Volume 45, 2015, Pages 76-85, ISSN 1877-0509,
https://doi.org/10.1016/j.procs.2015.03.090
(https://www.sciencedirect.com/science/article/pii/S1877050915003269)

[4] Jaychandra Reddy, V., & Nagalakshmi, T. J. (2019). Skin disease detection using artificial neural network. Indian Journal of Public Health Research and Development, 10(11), 3829–3832. https://www.ijitee.org/wp-content/uploads/papers/v8i12/L39481081219.pdf

[5] Anurag Kumar Verma, Saurabh Pal, Surjeet Kumar, Comparison of skin disease prediction by feature selection using ensemble data mining techniques, Informatics in Medicine Unlocked, Volume 16, 2019, 100202, ISSN 2352-9148, https://doi.org/10.1016/j.imu.2019.100202
(https://www.sciencedirect.com/science/article/pii/S2352914819300838)

[6] Singhal, E., & Tiwari, S. (2017). Skin Cancer Detection using Arificial Neural Network. *International Journal of Advanced Research in Computer Science*, *6*(1), 149-157. doi:https://doi.org/10.26483/ijarcs.v6i1.2402

[7] Hashmani MA, Jameel SM, Rizvi SSH, Shukla S. An Adaptive Federated Machine Learning-Based Intelligent System for Skin Disease Detection: A Step toward an Intelligent Dermoscopy Device. Applied Sciences. 2021; 11(5):2145. https://doi.org/10.3390/app11052145

[8] Shashi Rekha G, Prof. H. Srinivasa Murthy, Dr. Suderson Jena et al. “Digital Dermatology – skin Disease Detection Model Using Image Processing. Published in International Journal of

Innovative Research in Science, Engineering and Technology. Vol 7, Issue 7, July 2018. http://www.ijirset.com/upload/2018/july/7_1_DIGITAL.pdf

[9] Ambad, Pravin & Shirsat, A.. (2016). A Image analysis System to Detect Skin Diseases. IOSR Journal of VLSI and Signal Processing. 06. 17-25. 10.9790/4200-0605011725.
https://www.researchgate.net/publication/326753284_A_Image_analysis_System_to_Detect_Skin_Diseases

[10] Wei LS, Gan Q, Ji T. Skin Disease Recognition Method Based on Image Color and Texture Features. Comput Math Methods Med. 2018 Aug 26;2018:8145713. doi: 10.1155/2018/8145713. PMID: 30224935; PMCID: PMC6129338. https://pubmed.ncbi.nlm.nih.gov/30224935/

[11] R. Yasir, M. S. I. Nibir, and N. Ahmed, “A skin disease detection system for financially unstable people in developing countries,” Global Science and Technology Journal, vol. 3, no. 1, pp. 77–93, 2015.
https://www.academia.edu/20142809/A_Skin_Disease_Detection_System_for_Financially_Unstable_People_in_Developing_Countries

[12] Bhadula, S., Sharma, S., Juyal, P., & Kulshrestha, C. (2019). Machine Learning Algorithms based Skin Disease Detection. International Journal of Innovative Technology and Exploring Engineering, 9(2), 4044–4049. https://doi.org/10.35940/ijitee.b7686.129219

[13] R. Yasir, M. A. Rahman and N. Ahmed, "Dermatological disease detection using image processing and artificial neural network," 8th International Conference on Electrical and Computer Engineering, 2014, pp. 687-690, doi: 10.1109/ICECE.2014.7026918.
https://ieeexplore.ieee.org/abstract/document/7026918

[14] El Abbadi, N. K., Dahir, N. S., AL-Dhalimi, M. A. & Restom, H. (2010). Psoriasis Detection Using Skin Color and Texture Features. Journal of Computer Science, 6(6), 648-652. https://doi.org/10.3844/jcssp.2010.648.652

[15] Suneel Kumar and Ajit Singh. Image Processing for Recognition of Skin Diseases. *International Journal of Computer Applications* 149(3):37-40, September 2016. https://www.ijcaonline.org/archives/volume149/number3/25980-2016911373

[16] Połap D, Winnicka A, Serwata K, Kęsik K, Woźniak M. An Intelligent System for Monitoring Skin Diseases. *Sensors (Basel)*. 2018;18(8):2552. Published 2018 Aug 4. doi:10.3390/s18082552 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6111999/

[17] Mahamudul Hasan, Surajit Das Barman, Samia Islam, and Ahmed Wasif Reza. 2019. Skin Cancer Detection Using Convolutional Neural Network. In Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence (ICCAI '19). Association

for Computing Machinery, New York, NY, USA, 254–258. https://doi.org/10.1145/3330482.3330525

[18] M. Shamsul Arifin, M. Golam Kibria, A. Firoze, M. Ashraful Amini and Hong Yan, "Dermatological disease diagnosis using color-skin images," 2012 International Conference on Machine Learning and Cybernetics, 2012, pp. 1675-1680, doi: 10.1109/ICMLC.2012.6359626. https://ieeexplore.ieee.org/abstract/document/6359626

[19] V.R. Balaji, S.T. Suganthi, R. Rajadevi, V. Krishna Kumar, B. Saravana Balaji, Sanjeevi Pandiyan, Skin disease detection and segmentation using dynamic graph cut algorithm and classification through Naive Bayes classifier, Measurement, Volume 163, 2020, 107922, ISSN 0263-2241,
https://doi.org/10.1016/j.measurement.2020.107922
(https://www.sciencedirect.com/science/article/pii/S0263224120304607)

[20] https://datahub.io/machine-learning/dermatology

Appendix: A- Packages, Tools used & Working Process

Python Programming language

Python is a high-level Interpreter based programming language used especially for general-purpose programming. Python features a dynamic type of system and supports automatic memory management.

It supports multiple programming paradigms, including object-oriented, functional and Procedural and also has its a large and comprehensive standard library. Python is of two versions. They are Python 2 and Python 3.

This project uses the latest version of Python, i.e., Python 3. This python language uses different types of memory management techniques such as reference counting and a cycle-detecting garbage collector for memory management. One of its features is late binding (dynamic name resolution), which binds method and variable names during program execution.

Python's offers a design that supports some of things that are used for functional programming in the Lisp tradition. It has vast usage of functions for faster results such as filter, map, split, list comprehensions, dictionaries, sets and expressions. The standard library of python language has two modules like itertools and functools that implement functional tools taken from Standard machine learning.

Libraries

NumPy

Numpy is the basic package for scientific calculations and computations used along with Python. NumPy was created in 2005 by Travis Oliphant. It is open source so can be used freely. NumPy stands for Numerical Python. And it is used for working with arrays and mathematical computations.

Using NumPy in Python gives you much more functional behavior comparable to MATLAB because they both are interpreted, and they both allows the users to quickly write fast programs as far as most of the operations work on arrays, matrices instead of scalars. Numpy is a library consisting of array objects and a collection of those routines for processing those arrays.

Numpy has also functions that mostly works upon linear algebra, Fourier transform, arrays and matrices. In general scenario the working of numpy in the code involves searching, join, split, reshaping etc. operations using numpy.

The syntax for importing the numpy package is → import numpy as np indicates numpy is imported alias np.

Pandas

Pandas is used whenever working with matrix data, time series data and mostly on tabular data. Pandas is also open-source library which provides high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

This helps extremely in handling large amounts of data with help of data structures like Series, Data Frames etc. It has inbuilt methods for manipulating data in different formats like csv, html etc.,

Simply we can define pandas is used for data analysis and data manipulation and extremely works with data frames objects in our project, where data frame is dedicated structure for two-dimensional data, and it consists of rows and columns similar to database tables and excel spreadsheets.

In our code we firstly import pandas package alias pd and use pd in order to read the csv file and assign to data frame and in the further steps. We work on the data frames by manipulating them and we perform data cleaning by using functions on the data frames such as df.isna().sum(). So, finally the whole code depends on the data frames which are to be acquired by coordinating with pandas. So, this package plays a key role in our project.

Matplotlib

Matplotlib is a library used for plotting in the Python programming language and it is a numerical mathematical extension of NumPy. Matplotlib is most commonly used for visualization and data exploration in a way that statistics can be known clearly using different visual structures by creating the basic graphs like bar plots, scatter plots, histograms etc.

Matplotlib is a foundation for every visualizing library and the library also offers a great flexibility with regards to formatting and styling plots. We can choose freely certain assumptions like ways to display labels, grids, legends etc.

In our code firstly we import the matplotlib.pyplot alias plt, This plt comes into picture in the exploratory data analysis part to analyze and summarize datasets into visual methods, we use plt to add some characteristics to figures such as title, legends, labels on x and y axis as said earlier ,to understand more clearly we can also use different plots.

Seaborn

Seaborn is used for drawing attractive statistical graphics with just a few lines of code. In other words, we can say seaborn is a data visualization library based on the matplotlib and closely combined with Pandas data structures in Python. Visualization is the central theme of Seaborn which helps in exploration and understanding of data.

Plots are used for visualizing the relationship between variables. Those variables can be numerical or categorical.

Using Seaborn, we can also plot wide varieties of plots like:

1. Distribution plots
2. Pie chart and bar chart
3. Scatter plots
4. Pair plots
5. Heat maps

In our code we use seaborn library in EDA where sns is used to create countplot between skin disease and count of target values and we use facetgrid with respect to sns for looking out distribution of age based on the diseases and sns with respect to the countplot is also used to find perspective of data analysis i.e., is the disease due to family genes i.e., family history vs count. As said earlier, sns also used in determining the heatmap.

Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. Scikit learn is an efficient, and beginner, user friendly tool for predictive data analysis and it provides a selection of tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library is built upon different libraries such as NumPy, SciPy and Matplotlib.

Scikit learn is used when identifying to which category an object likely belongs, predicting continuous values and grouping of similar objects into clusters.

When coming to our code, sklearn plays an important with respect to classification algorithm, the final result and performance. Accuracy of the algorithm can be determined using sklearn. The different modules imported from sklearn library is train_test_split, GaussianNB, one vs rest classifier and all the metrics. When going much detail into the modules and packages, train_test_split means it splits the data into random training and testing subsets. We used gaussian naive bayes classification algorithm in order to classify the values of the model. On taking the syntax as example from sklearn.metrics, import * describes to import all the metrics required for doing some kind of mathematical, or evidential calculations. Similarly from sklearn.model_selection, import train_test_split described above and there are few preprocessing steps such as from sklearn.preprocessing import LabelEncoder where labelencoder encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels, and other step is from sklearn.preprocessing import label_binarize where label binarize is used to convert multi-class labels to binary labels (belong or does not belong to the class) and we use multiclass classification in our which will be explained in detail in the above document, and when coming to metrics we use confusion matrix in order to calculate the performance of classification model by using certain measures like precision,recall,f1 sore and threshold value.

Supervised Learning algorithms − Supervised learning is one of the machine learning approaches through which models are trained using perfectly labelled training data and on the basis of that models predicts the output. Almost all the popularly known supervised learning algorithms, Such as Linear Regression, Support Vector Machine (SVM), Decision Tree, Naïve bayes etc., are the part of scikit-learn.

Unsupervised Learning algorithms − Unsupervised learning is also one of the machine learning approaches, through which models are not supervised using training data. Instead of that model itself finds the hidden patterns and insights from the given data.

On the other side it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

Clustering − This model can be used to group unlabelled data.
Cross Validation −This process is used to check accuracy on unseen data in supervised models.
Dimensionality Reduction – Dimensionalities are nothing but attributes of the data. This step helps in reducing the number of attributes in data which can be used further for tasks like feature selection, visualization and summarization.
Ensemble methods – Ensemble means to combine. These methods combine various predictions of multiple supervised models.
Feature extraction – This step is used to define attributes by extracting the features from the dataset having data of any form.
Feature selection – The extracted features contain lots of features some of which may be not useful. Feature selection is the process of identifying the important features for the creation of supervised models.
Open Source − It is an open-source library and also commercially usable under BSD license.
Tools Used Jupyter Notebooks:

Jupyter Project is spin-off project from the I-Python project, which is initially provided an interface only for the Python language. The name Jupyter is derived from the combination of Julia, Python, and R.

A Jupyter Notebook is basically a JSON file with a number of annotations. There are mainly three parts of the Notebook as follows.

● **Metadata:** A data dictionary of definitions used to set-up and display the notebook,it is alos be said as data about data.

● **Notebook format:** the Version numbers of the software used to create the notebook. The version number is used for backward compatibility**.**

● **List of cells:** there are three different types of different cells listed beside — markdown (display), code (to excite), and output.

Appendix: B- Source Code

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('whitegrid')
import vpython as vp
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import *
df=pd.read_csv('dermatology.csv',header=None)
df.info()
df.isna().sum()
sum(df.duplicated())
df.describe()
newNames=['erythema','scaling','definite_borders','itching','koebner_phenomenon','polygonal_papules','follicular_papules','oral_mucosal_involvement','knee_elbow_involvement','scalp_involvement','family_history','melanin','eosinophilis','PNL','fibrosis','exocytosis','acanthosis','hyperkeratosis','parakeratosis','clubbing','elongation','suprapapillary_epidermis','spongiform_pustule','munro_microabcess','focal_hypergranulosis','granular_layer','basal_layer','spongiosis','sawtooth','follicular','perifollicular','inflammatory_monoluclear','band-like
infiltrate','Age','target']
for i in range(35):
   df.rename(columns={i:newNames[i]},inplace=True)
df.head()
df.shape
df=df.loc[df.Age!='?']
```

```
df.shape
df.Age=df.Age.astype('int64')
df.info()
group_map={1:'psoriasis',2:'seboreic_dermatitis',3:'lichen_planus',4:'pityriasis_rosea',5:'cronic_dermatitis',6:'pityriasis_rubra'}
df['target']=df['target'].map(group_map)
plt.figure(figsize=(10,5))
sns.countplot(df['target'])
plt.xticks(rotation=45,fontsize=13)
plt.title('Skin Disease Vs Count',fontsize=14)
plt.ylabel('Count',fontsize=13)
plt.xlabel('Skin Disease',fontsize=13)
facet=sns.FacetGrid(df,hue="target",aspect=4)
facet.map(sns.kdeplot,'Age',shade=False).add_legend()
plt.title("Distribution of Age According to Skin Disease")
sns.FacetGrid(df,hue='target',aspect=3,margin_titles=True).map(sns.kdeplot,'itching',shade=True).add_legend()
plt.title('Skin Disease Vs Itching Factor')
print(df.family_history.value_counts())
sns.countplot(df.family_history)
plt.title('Family History Vs Count')
family_h = df.loc[df.family_history==1]
family_nh = df.loc[df.family_history==0]
fig = plt.figure(figsize=(16,5))
plt.subplot(121)
sns.countplot(family_h['target'])
plt.xticks(rotation=45,fontsize=13)
plt.title('Family History: Yes',fontsize=14)
plt.ylabel('Count',fontsize=13)
plt.xlabel('Skin Disease',fontsize=13)
plt.subplot(122)
sns.countplot(family_nh['target'])
plt.title('Family History: No',fontsize=14)
plt.xticks(rotation=45,fontsize=13)
plt.ylabel('Count',fontsize=13)
plt.xlabel('Skin Disease',fontsize=13)
fig = plt.figure(figsize=(18, 18))
sns.heatmap(df.iloc[:,2:].corr(),cmap="RdYlGn",annot=True, linewidths=.5, fmt= '.1f')
plt.xticks(fontsize=11,rotation=70)
plt.show()
for element in newNames:
    if (element != "Age")and (element != "family_history") and (element != "target"):
        df.loc[df.loc[:, element]== 0, element] = "absent"
        df.loc[df.loc[:, element]== 1, element] = "Low"
        df.loc[df.loc[:, element]== 2, element] = "Medium"
        df.loc[df.loc[:, element]== 3, element] = "High"
    elif (element == "family_history"):
        df.loc[df.loc[:, element] == 0, element] = "NO"
        df.loc[df.loc[:, element] == 1, element] = "YES"
    elif (element == "target"):
        df.loc[df.loc[:, element]== 1, element] = "target[0]"
        df.loc[df.loc[:, element]== 2, element] = "target[1]"
        df.loc[df.loc[:, element]== 3, element] = "target[2]"
        df.loc[df.loc[:, element]== 4, element] = "target[3]"
        df.loc[df.loc[:, element]== 5, element] = "target[4]"
        df.loc[df.loc[:, element]== 6, element] = "target[5]"
else:
     pass
#checking the number of data in each category to determine possibility of consolidation
for element in newNames:
    if element != "Age":
        print(df.loc[:, element].value_counts())
df.loc[df.loc[:, "band-like infiltrate"] == "Low", "band-like infiltrate"] = "absent"
df.loc[df.loc[:, "perifollicular"] == "Low", "perifollicular"] = "Medium"
df.loc[df.loc[:, "parakeratosis"] == "Low", "parakeratosis"] = "Medium"
df.loc[df.loc[:, "follicular"] == "Medium", "follicular"] = "Low"
df.loc[df.loc[:, "sawtooth"] == "Low", "sawtooth"] = "Medium"
df.loc[df.loc[:, "spongiosis"] == "Low", "spongiosis"] = "Medium"
df.loc[df.loc[:, "fibrosis"] == "Low", "fibrosis"] = "Medium"
df.loc[df.loc[:, "eosinophilis"] == "Medium","eosinophilis" ] = "Low"
df.loc[df.loc[:, "melanin"] == "Low", "melanin"] = "Medium"
df.loc[df.loc[:, "oral_mucosal_involvement"] == "Low", "oral_mucosal_involvement"] = "Medium"
df.loc[df.loc[:, "polygonal_papules"] == "Low", "polygonal_papules"] = "absent"
for i in newNames:
    if (i != "Age") and (i != "target") and (i != "family_history"):
        for element in df.loc[:, i].unique():
            df.loc[:, str(i) + str(element)] = (df.loc[:, i] == element).astype(int)
df.shape
for i in newNames:
    if (i != "Age") and (i != "target") and (i != "family_history"):
        df = df.drop(i, axis = 1)
df.shape
#Encodes the categorical variables in diagnosis
labelencoder = LabelEncoder()
for i in newNames:
    if (i == "target") or (i == "family_history"):
        df.loc[:, i] = labelencoder.fit_transform(df.loc[:, i])
#changes the index of outcome column
df = df.reindex(list([a for a in df.columns if a != 'target'] + ['target']), axis=1)
n = len(df.columns) #determines the number of columns in dataset
X = df.iloc[:, :(n-1)].values
Y = df.iloc[:, (n-1)].values
y = label_binarize(Y, classes=[0, 1, 2, 3, 4, 5]) #binerize target column for calculating fpr, tpr and threshold
n_classes = y.shape[1]
#shuffle and split training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
classifier = OneVsRestClassifier(GaussianNB()) #predict each class against the other
```

```
classifier.fit(X_train, y_train)
y_predict = classifier.predict(X_test)
probas = classifier.predict_proba(X_test)
#compute ROC Curve for each class
fpr=dict()
tpr=dict()
roc_auc=dict()
th=dict()
CM = dict()
P = dict()
R = dict ()
F1 = dict()
for i in range(n_classes):
    fpr[i],tpr[i],th[i]=roc_curve(y_test[;,i],probas[:,i])
    roc_auc[i]=auc(fpr[i],tpr[i])
    CM[i] = confusion_matrix (y_test[:, i], y_predict[:, i])
    print ('Confusion matrix (GaussianNB) for class', i, 'vs other classes: \n', CM[i])
    print("GaussianNB | Probability Threshold for class",i,"\n",th[i])
    P[i] = precision_score(y_test[:, i], y_predict[:, i])
    print ("\nPrecision:", np.round(P[i], 3))
    R[i] = recall_score(y_test[:, i], y_predict[:, i])
    print ("\nRecall:", np.round(R[i], 3))
    F1[i] = f1_score(y_test[:, i], y_predict[:, i])
    print ("\nF1 score:", np.round(F1[i], 3))
plt.figure()
lw=2 #line width
color=['b','r','c','y','m','k']
for i in range(n_classes):

plt.plot(fpr[i],tpr[i],color=color[i],\lw=lw,label="ROCcurveofclass=%s"%(i,np.round(roc_auc[i],3)),linestyle=":")
    plt.plot([0,1],[0,1],color="navy",lw=lw,linestyle="--")
    plt.xlim([0.0,1.0])
    plt.ylim([0.0,1.05])
    plt.xlabel("False positive rate")
    plt.ylabel("True positive rate")
    plt.title("Receiver operating characteristic of GaussianNB")
    plt.legend(loc="lower right")
    plt.show()
```