# Advanced Machine Learning Techniques for Personalized Alopecia Areata Intervention Modeling

by

MUTASIM FOUAD SHOWMIK
21101052
MOHAMMED RUHAN
21101042
MD MOZAHIDUL ISLAM
21301487
RAFIA SULTANA
21101085
SADIA ZABIN RISHTA
21301043

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
January 2025

# Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

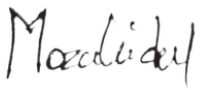
**Student's Full Name & Signature:**

MUTASIM FOUAD SHOWMIK

21101052

MOHAMMED RUHAN

21101042

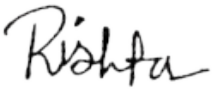


MD MOZAHIDUL ISLAM

21301487

RAFIA SULTANA

21101085



SADIA ZABIN RISHTA

21301043

# Approval

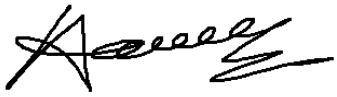
The thesis titled "Advanced Machine Learning Techniques for Personalized Alopecia Intervention Modeling" submitted by

1. MUTASIM FOUAD SHOWMIK (21101052)
2. MOHAMMED RUHAN (21101042)
3. MD MOZAHIDUL ISLAM (21301487)
4. RAFIA SULTANA (21101085)
5. SADIA ZABIN RISHTA (21301043)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 2025.

**Examining Committee:**

Supervisor:
(Member)



Amitabha Chakrabarty, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

# Abstract

Alopecia Areata (AA) is a skin condition that causes hair loss, ranging from small patches to complete baldness. Diagnosis and treatment are not very easy because of the complex interaction among genetic, autoimmune, hormonal, environmental, and sociodemographics. In this study, machine learning techniques were explored to assess different stages of Alopecia Areata to allow early diagnosis and treatment. Three complementary datasets consisting of clinical, genetic, environmental, and demographic features were used to build and validate predictive models. Several machine learning models were trained, including Meta Classifiers, K-nearest neighbors (KNNs), and Random Forest models. Among all the models tested, the Meta Classifier achieved the highest accuracy and was selected as the base model for feature extraction by LIME and SHAP. Identified key features impacting the models were fed into the fusion datasets for having advanced models such as 1D CNN with autoencoder. The best-performing model was the 1D CNN with an autoencoder, with a 96.78% accuracy, able to integrate the features extracted. This study shows the capability of machine learning to aid early diagnosis and personalized management of alopecia areata and subsequently provides an avenue for better and more reliable intervention strategies.



**Keywords:** Alopecia Areata (AA),, Diagnosis, Clinical, Sociodemographic, Feature Extraction, K-Nearest Neighbors (KNN), Meta Classifiers, Random Forest, LIME, SHAP, 1D CNN with Autoencoder

# Acknowledgement

We would like to express our gratitude to Almighty Allah for providing us with the strength of both physically and mentally to complete our thesis titled "Advanced Machine Learning Techniques for Personalised Alopecia Intervention Modeling" within the stipulated period.

We express our gratitude to our supervisor, Dr. Amitabha Chakrabarty, Professor of Computer Science and Engineering. Throughout the process of writing my thesis, his advice, support, supervision, motivation, and suggestions were very helpful. We like to convey that his steadfast support enabled us to thoroughly develop our work and concepts. The outcomes of this study were successful due to both individual contributions and the collective efforts of many others.

Without the full support of our Department of Computer Science, this study would not have been possible. We want to thank our parents for always being there for us and supporting us as we've built our careers. Their help was very important in finishing our thesis work.

# Table of Contents

# List of Figures

# List of Tables

# Nomenclature

The following list describes several symbols & abbreviation that will be later used within the body of the document

$AA$ Alopecia Areata

$ADASYN$ A Machine Learning Algorithm that Creates Synthetic Data to Balance Imbalanced Datasets

$AUC$ Area Under Curve

$BCE$ Binary Cross Entropy

$CNN$ Convolutional Neural Network

$DT$ Decision Tree

$FCNN$ Fully Connected Neural Network

$FPHL$ Female Pattern Hair Loss

$FRCNN$ Faster Residual Convolutional Neural Network

$KNN$ K-Nearest Neighbors

$LASSO$ Least Absolute Shrinkage and Selection Operator

$LIME$ Local Interpretable Model-agnostic Explanations

$MSE$ Mean Squared Error

$MSVM$ Multi-Class Support Vector Machine

$PCOS$ Polycystic Ovary Syndrome

$RBF$ Radial Basis Function

$RCNN$ Region-Based Convolutional Neural Networks

$RFC$ Random Forest Classifier

$ROC$ Receiver Operating Characteristic

$SHAP$ SHapley Additive exPlanations

$SMOTE$ Synthetic Minority Over-sampling Technique

$ssGSEA$ STEM Algorithm Single Sample Gene Set Enrichment Analysis

$SVC$ Support Vector Classifier

$SVM$ Support Vector Machine

$SVM-RFE$ Support Vector Machine-Recursive Feature Elimination

$WGCNA$ Weighted Gene Co-Expression Network Analysis

$XGB$ XGBoost Classifier

# Chapter 1

# Introduction

Alopecia Areata, most commonly known as baldness, is a worldwide problem that is affecting millions of people and causing them considerable emotional distress, leading to diminished self-esteem. This condition can affect both men and women. In men, symptoms include gradual thinning of hair or complete baldness, depending on age, genetics, hormonal imbalance, and certain medical conditions. Although one of the most common forms of hair loss, it occupies its place in women too. Its severity ranges from minor thinning to total baldness which is usually dependent upon factors such as hormonal changes, malnutrition, thyroid disorders, and in some cases, Polycystic Ovary Syndrome(PCOS). Recent data show increasing sizes of the populace is getting afflicted with alopecia worldwide, for example, in South Korea where cases rose from 103,000 in 2001 to 233,000 in 2020. Because traditional diagnosis and treatments usually involve visitations to specialized scalp clinics, they tend to be costly in comparison to others, hence sometimes inaccessible by more individuals. Therefore, this move seeks to develop more room for otherwise cost-prohibitive, accessible, home-based diagnosis methods. Our research makes use of machine learning methods in diagnosing the various stages of alopecia and treatments with the likelihood of curing it with the application of numerical data. The basis for our computations is going to be provision of patient demographics, medical history, and biochemical markers, but not on the basis of image data. As far as feature extraction is concerned, advanced methods such as LIME and SHAP could be used to showcase those variables that are contributing majorly to machine learning performance. With these characteristics extracted and later incorporated into the models, it helps to offer better accuracy and reliability on predicting its performance. The aim of the entire effort is to develop a home tool for the early detection and continuity of alopecia enabling individuals to have hands-on experience in the supervision of their hair. Users should be able to input their numerical data into machine learning and get suggestions on alopecia stage and treatment. This study presents numerical data and feature analysis with the probability to disrupt alopecia management, decreasing dependence on the classic clinical visits and focusing on the roles of impactful features in increasing model performance.

## 1.1 Problem Statement

Alopecia is a common illness marked by hair loss which leads to a great deal of social and psychological pain. Although specific techniques have been suggested for deal-

ing with distinct alopecia syndromes, success remains rarely moderate. Customized interventions are necessary to be employed for the diverse and particular displays of alopecia areata. The main challenge is to devise advanced machine learning techniques that can accommodate all intricate relationships between features including genetic susceptibility, environmental exposure, and treatment response necessary for the creation of effective and unique interventions. Past studies have discussed various machine learning algorithms for the prediction of alopecia areata; nevertheless, several of the models were not quite suitable for this purpose, especially when predicting the actual application. Possible drawbacks in predicting effectiveness are due to a lack of accurate methods for extracting features from the inputs. This research aims to bridge this gap, by implementation of novel machine learning techniques and advanced feature extraction to produce more accurate, personalized models for alopecia prediction and management.

## 1.2 Aims and Objectives

Our study focuses on developing an integrated machine learning system which effectively learns from numerical datasets alongside extracted features to improve Alopecia intervention accuracy and personalization. The actual understanding of alopecia as a repetitive dermatologic disorder that causes physical and emotional distress has motivated this study to build better diagnostic tools together with customized treatment options and patient management methodology. To achieve this, we propose a multimodal learning framework designed to predict and personalize alopecia interventions by leveraging the complementary insights derived from numerical data and extracted features.The research will begin by compiling three distinct datasets. The study utilizes three datasets comprising numerical patient information alongside demographic and medical background and biochemical markers along with feature databases containing specific extracted information. Analysis of these datasets will help train and optimize sophisticated machine learning models that extract important features which enhance prediction performance for hair loss. A single dataset will merge extracted features and numerical data to enhance predictive accuracy and reliability of the models.

# Chapter 2

# Literature Review

Kapoor and Mishra (2018) [9] focused on the early diagnosis of forms of alopecia using feed-forward neural networks and back-propagation as a means to classify affected and unaffected individuals with alopecia. Kapoor and Mishra 2018 [9] claimed that it has high accuracy (91%) and consider the system beneficial in areas where medical professionals are sparse altogether. The accuracy of the present program itself depends on the quality of data, and this reflects a general issue with machine learning applications in terms of data availability and bias, which cannot be neglected in practical environments. In contrast, Madke et al. (2023) [26] use various machine learning approaches, including KNN and SVM classifiers, for classifying alopecia severity in men. Hair loss is categorized into four classes using a fairly large database from Daegu University and anonymous sources for better robustness in classification.

Vujovic and Del Marmol (2014) [7] present a thorough review of Female Pattern Hair Loss (FPHL), dealing foremost with its origin, diagnostics, and treatment, without directly applying machine learning. Their work will serve as a preliminary comparison with studies like Kapoor and Mishra (2018) [9] and Madke et al. (2023) [26], laying further emphasis on requiring machine learning models to involve diverse data about both male and female patterns of alopecia for greater generalizability.

Kim et al. (2022) [22] put advanced object detection, EfficientDet, and YOLOv4, to work in detecting scalp hair follicles from images, which would be of great help in assessing hair transplantation areas. In terms of image data impact on hair loss analysis, it is quite similar to Madke et al. (2023) [26] but focuses even more on the distinguishing characteristics of new algorithms allowing for accurate measurements, which are crucial to transplant planning; the importance of data augmentation to avoid over-fitting is another important point where agreement is noticed across machine learning applications, such as the limitations discussed by Kapoor and Mishra (2018) [9].

Saraswathi and Pushpa (2023) [28] used a Faster Residual Convolutional Neural Network(FRCNN) to diagnose Alopecia Areata and more scalp conditions with an accuracy of 84.3%. This study is comparable to that of Kim et al. (2022) [22], in using deep learning for image analysis, but is set apart in its focus on a wider range of scalp conditions. The accuracy achieved indicates that deep learning can enhance diagnostics, but similar to other studies, it also points to the associated challenges of high computational requirements as well as the need for large, well-labeled training datasets.

Kim et al. (2022) [22] address possible solutions by normalizing and carrying out geometric transforms to added data variety somehow trying to clear from bias towards mild alopecia cases leaning more to their dataset. They have suggested methods such as SMOTE and ADASYN for preparing an appropriate balance, and propose putting together with medical institutions to create a more diversified dataset. This further emphasizes the need for good datasets that are crucial in bringing effective diagnostic tools, also reflecting the challenges dealt with by Sayyad and Midhunchakkaravarthy (2022) [23] and Sayyad et al. (2022) [19] that also underscore the importance of data quality and feature selection in model performance.

Sayyad and Midhunchakkaravarthy (2022) [23] present an application of SVM and KNN classifiers for diagnosing Alopecia Areata with a system they called ScalpEye Their approach emphasizes the extraction of meaningful features from facial images, a methodology that relies heavily on the manipulation of data because incorrect handling of data may lead to features being misrepresented. Sayyad et al. (2022) [19] also raise a concern regarding feature extraction, although the method of data processing includes the application of histogram equalization and data augmentation to improve their classification accuracy; they utilize VGG-19 for feature extraction and SVM for classification.

Khatun et al. (2022) [20] analyzed hair fall using the machine learning models with an emphasis on the Bangladeshi subcontinent, citing possible aspects of gender bias and the necessity for a dataset that covers a wider age group and geographic representation as the complications. Their insistence on rigid data quality checks and a pool of preprocessing methods resonates with the overarching agenda in these studies, which is the improvement of data integrity for reliable models.

With a focus on Alopecia Areata prediction, Aditya et al. (2022) [18] take an overview of a much wider angle, including the collection of large datasets and the use of various machine learning processes, such as CNN. They are insistent on using diverse data sources in the training of robust and generalizable models, which in itself presents the already-encountered current challenge in this field of research; building good-quality comprehensive datasets that can accurately reflect population diversity.

Roy and Protity (2022) [27] and Saraswathi and Pushpa (2023) [29] have both reached a deep learning solution for the diagnostic tool, one further differentiating on the aspects of scalability and applicability. With a two-layer feed-forward neural network and high accuracy considered, Roy and Protity (2022) [27] emphasize image preprocessing and data balancing, whereas Saraswathi and Pushpa (2023) [29] implement a Multi-Class SVM to address different severity levels of scalp conditions, demonstrating SVM's capability to deal with challenging classification problems.

Models are by nature complex, thus making it difficult for any external agent to explain their behavior. Surely, it is one of such difficulties that made Heng (2023) [24] focus on techniques such as LIME and Occlusion Sensitivity to enable the enhancement of AI model explainability in biomedical applications, with particular reference to those models used for image analysis. Such methods help to map out which image areas are considered by the model to be more or less responsible for a certain prediction, thereby lending more credence to the trustworthiness of these models or the confidence of the users towards such methods. Transparency is an important aspect when models are used in decision-making situations, particularly in health settings, where the ability to explain decision pathways could either make

or break a diagnostic situation.
Deep learning frameworks were employed by both Benhabiles et al. (2019) [10] and Lee et al. (2020) [13] but with different focus considerations. While Benhabiles et al [3] were evaluating various CNN architecture efficiencies in hair loss pattern detection from facial images, they now particularly point to the DEX-IMDB-WIKI architecture for its highest performance, emphasizing the promise of CNN in wider contexts other than those of healthcare, such as security and commercial interests. On the contrary, Lee et al. (2020) [13] assess hair loss in patients with alopecia areata using deep learning to enhance predictive accuracy and reproducibility in hair loss assessment. These studies addressed practical implementation issues such as requiring very little image preprocessing and enhancing inter-rater reliability.
Behal et al. (2024) [31] utilized CNN models to identify stages of hair loss, employing image processing techniques that classify stages with accuracy based on the Hamilton-Norwood scale. The aim is to minimize the time taken for diagnosis to facilitate a more accurate treatment plan. It is thus conceivable to conclude that deep learning can facilitate the very practical outcome of adding value to human life by providing a quicker and more accurate diagnosis.
the work of Xiong et al. (2023) [30] centered on the genetic basis for severe alopecia areata and employs machine learning to analyze gene expression data to identify genes associated with immune monitoring. An integration of bioinformatics and machine-learning techniques exemplifies how diagnostic tools can be developed to predict disease severity and treatment response, which is a big step toward personalized medicine.
In 2017, Chang et al. [12] presented 'ScalpEye,' a system that uses SSD Inception v2, Faster R-CNN Inception v2, and Faster R-CNN Inception ResNet v2 Atrous to analyze commonly found scalp conditions from microscopic images, such as dandruff, folliculitis, hair loss, and oiliness. In a similar way, Kim et al. (2022) [21], through multitask learning using Mask R-CNN, were able to detect hair follicles and measure the levels of hair loss accurately. Such findings additionally indicate that deep learning models can be applied to classify hair follicle health status into healthy, normal, or severe categories, thereby assisting in clinical determination.
Chang et al. [12] and Kim et al. [21] analyze the capability of deep learning models to analyze and interpret image data for diagnosing conditions affecting hair and scalp, together with the precision, recall, and accuracy of those models. The studies together argue that AI can assist in minimizing the chances for diagnostic errors, augmenting the training process, and perhaps decrease the costs associated with health care through automating assessments of complex diagnoses.
Alopecia areata as a condition dealt with ferroptosis and its biomarkers "SLC40A1", "LCN2", "CREB5", and "SLC7A110" were put forward to be studied by Xu et al (2024)[32]. This study stated these biomarkers as significantly downregulated or less expressed in alopecia areata lesions as compared to normal scalp tissues and thus, considered to be promising for diagnostics. Machine learning, in other words, LASSO regression models and SVM-recursive feature elimination, was applied to trace these biomarkers from transcriptome data and were widely cited by Kim et al. [21], in using deep learning for image analysis.

## 2.1 Summary of Literature Review

| Ref.No. | Model | Dataset | Accuracy |
|---|---|---|---|
| [9] | Neural network with the backpropagation algorithm | Dataset for early diagnosis of alopecia | 91.0% |
| [10] | DEX-IMDB-WIKI network<br>DEX-ChaLearn Network | Head scalp image dataset | DEX-IMDB-WIKI network:85.5%<br>DEX-ChaLearn Network:84.5% |
| [12] | Faster-R-CNN-Inception-ResNetV2-Atrous | Scalp image dataset | Faster-R-CNN-Inception-ResNetV2-Atrous:90.9% |
| [13] | AloNet version 1.0 | 5 AloNet version 1.0 Clinical image dataset of alopecia areata | Jaccard similarity index: 94.1%<br>Hair loss area:96.3 % |
| [18] | Support Vector Machine (SVM)<br>Convolutional Neural Network (CNN)<br>K-Nearest Neighbors (KNN)<br>Random Forest Classifier<br>Gaussian Naive Bayes | Alopecia Areata Image Dataset<br>Healthy Hair Image Dataset | Support Vector Machine(SVM):85%<br>Convolutional Neural Network(CNN):92%<br>K-Nearest Neighbors (KNN):78%<br>RandomForestClassifier :88%<br>Gaussian Naive-Bayes:80%. |
| [19] | VGG-19 pre-trained CNN model<br>Support Vector Machine (SVM)<br>CNN-VGG-16 Model | Figaro1k,<br>Dermnet | VGG-19 pre-trained CNNmodel Support Vector Machine(SVM):90:12%<br>CNN-VGG-16Model:98.31% |
| [20] | XGBoost Classifier<br>Decision Tree<br>Random Forest<br>Support Vector Machine (SVM)<br>Logistic Regression<br>Naive Bayes<br>K-nearest Neighbor | Survey conducted among the Bangladeshi community | XGBoost Classifier: 92.62%<br>Decision Tree: 90.16%<br>Random Forest: 91.80%<br>Support Vector Machine: 86.89%<br>Logistic Regression: 85.25%<br>Naive Bayes: 79.51%<br>K-nearest Neighbor: 83% |

| Ref.No. | Model | Dataset | Accuracy |
|---|---|---|---|
| [21] | Mask R-CNN | Figaro1k<br>Dermnet | 79.29% |
| [22] | EfficientDet<br>YOLOv4<br>DetectoRS | National Information Society Agency Hair Scalp Dataset | EfficientDet: 64.71%<br>YOLOv4: 75.73%<br>DetectoRS: 66.36% |
| [23] | Support Vector Machine (SVM)<br>K-Nearest Neighbors (KNN)<br>Deep learning algo<br>Faster R-CNN by Inception ResNet v2 Atrous Model | Figaro1k<br>Dermnet | (SVM): 91.4%.<br>(KNN): 88.9%.<br>Faster R-CNN by Inception ResNet v2 Atrous Model: 97.41% to 99.09%. |
| [24] | Inception-v3<br>SqueezeNet RMSProp | Dermnet<br>Figaro1k | Inception-v3 with RMSProp: Training Accuracy: 98.4%<br>Validation Accuracy: 63.9%<br>SqueezeNet with RMSProp: Training Accuracy: 100.0%<br>Validation Accuracy: 100.0% |
| [25] | ResNet101<br>ResNeXt50<br>DenseNet169<br>XceptionNet41 | AI Hub | Ensemble of DenseNet, XceptionNet, and ResNet: Achieved the highest accuracy of 95.75% |
| [26] | K-Nearest Neighbors (KNN) classifier<br>Support Vector Machine (SVM) classifier | Dataset from Daegu University<br>Anonymously Sourced Dataset | K-Nearest Neighbors (KNN) classifier: 81%,<br>Support Vector Machine with a polynomial kernel (SVM Poly): 80%,<br>Support Vector Machine with a radial basis function kernel (SVM RBF): 78%. |
| [27] | convolutional Neural Network (CNN) | Hair and Scalp Disease Detection Dataset | 91.1% |
| [28] | Faster Residual Convolutional Neural Network (FRCNN) | Figaro1k,<br>Dermnet | 84.3% |

| Ref.No. | Model | Dataset | Accuracy |
|---|---|---|---|
| [29] | Multi-Class Support Vector Machine (SVM)<br>K-Nearest Neighbors (KNN)<br>HDM-Net<br>YOLOv4 | Figaro1k<br>Dermnet | K-Nearest Neighbors (KNN): 69.58%,<br>HDM-Net: 75.49%,<br>YOLOv4: 80.65%,<br>Multi-Class Support Vector Machine (MSVM): 87.68% |
| [30] | Weighted Gene Co-Expression Network Analysis (WGCNA)<br>Lasso Regression<br>Random Forest (RF)<br>STEM Algorithm<br>Single Sample Gene Set Enrichment Analysis (ssGSEA) | GSE68801<br>GSE45512<br>GSE80342 | LGR5: 88.5%<br>SHISA2: 89.9%<br>HOXC13: 90.0%<br>S100A3: 88.2% |
| [32] | Support Vector Machine-Recursive Feature Elimination (SVM-RFE<br>Least Absolute Shrinkage and Selection Operator (LASSO) regression model | GSE68801<br>GSE80342 | Support Vector Machine-Recursive Feature Elimination (SVM-RFE<br>Least GSE68801<br>GSE80342 90.5%,<br>Support Vector Machine-Recursive Feature Elimination :90.5%. |

Table 2.1: Summary of Related Works

## 2.2 Summary of Research Limitations

- The increasing body of literature on image processing and machine learning models for a variety of purposes, including medical imaging, has uncovered a number of notable weaknesses.
- A significant limitation is the common practice of only using one kind of dataset, namely graphical or numerical data, which ignores the potential advantages in combining different data types.
- In a siloed approach, researchers work on separate areas of problem space leading to models performing very well in the dataset the model has seen during training but failing to generalize more diverse or real-world scenarios.
- A further drawback is that the datasets are quite small in most of the studies. Additionally, the lack of interpretability in models like support vector machine(SVM) and region-based convolutional neural networks(RCNN) has continued as a significant issue.

- One shortcoming of machine learning approaches is that they rely heavily on large datasets with many examples of the trends one wishes to recognize.
- No proper description of the feature selection process and its effect on classification accuracy. Lack of details on parameters and training philosophy of the completed model requirements for validation in previously unseen datasets or real-world scenarios to perform generalization.

For the above limitations, we are going to perform the dataset concatenation and extracting out the impactful key features that can be the source of alopecia areata, which will be validated. Furthermore, the graphical and numerical dataset fusion will improve the accuracy of the models trained and tested.

# Chapter 3

# Methodology and Model Architecture

This section discusses our study's methodology, model architecture, and processes. It presents an elaborate description of the work plan and a workflow diagram illustrating the exact steps followed for deploying and assessing the model. To facilitate understanding of the design and functional aspects of the models used in the predictive analysis, the structure of the model architecture is also shown in this section, with the associated equations used by models.

## 3.1 Workflow

The process of preparing data for modeling begins with inputting datasets and goes on with the preprocessing steps combined with data transformation, data cleaning, and feature engineering. Several machine learning models such as Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree Classifier, Random Forest, XGB Classifier, and Support Vector Classifier (SVC) are trained and evaluated using a set of metrics such as accuracy, recall, f1-score, roc curve, and confusion matrix. To decide on the most predictive features, LIME and SHAP are used and the best scoring model will be deemed the basis for the feature importance analysis; moreover, SHAP was used with the help of the Pearson correlation matrix to extract impactful features. The most influential features were imputed across the three datasets to make a common feature set among the datasets. After that, a merging approach was implemented by using column consolidation with row-wise concatenation to initialize a merged dataset. After merging datasets the impact of merged and non-merged datasets is compared with the predictive models. Afterward, a tuned hybrid model of CNN with autoencoder was implemented to visualize the performance metrics over the merged dataset as shown in figure 3.1.

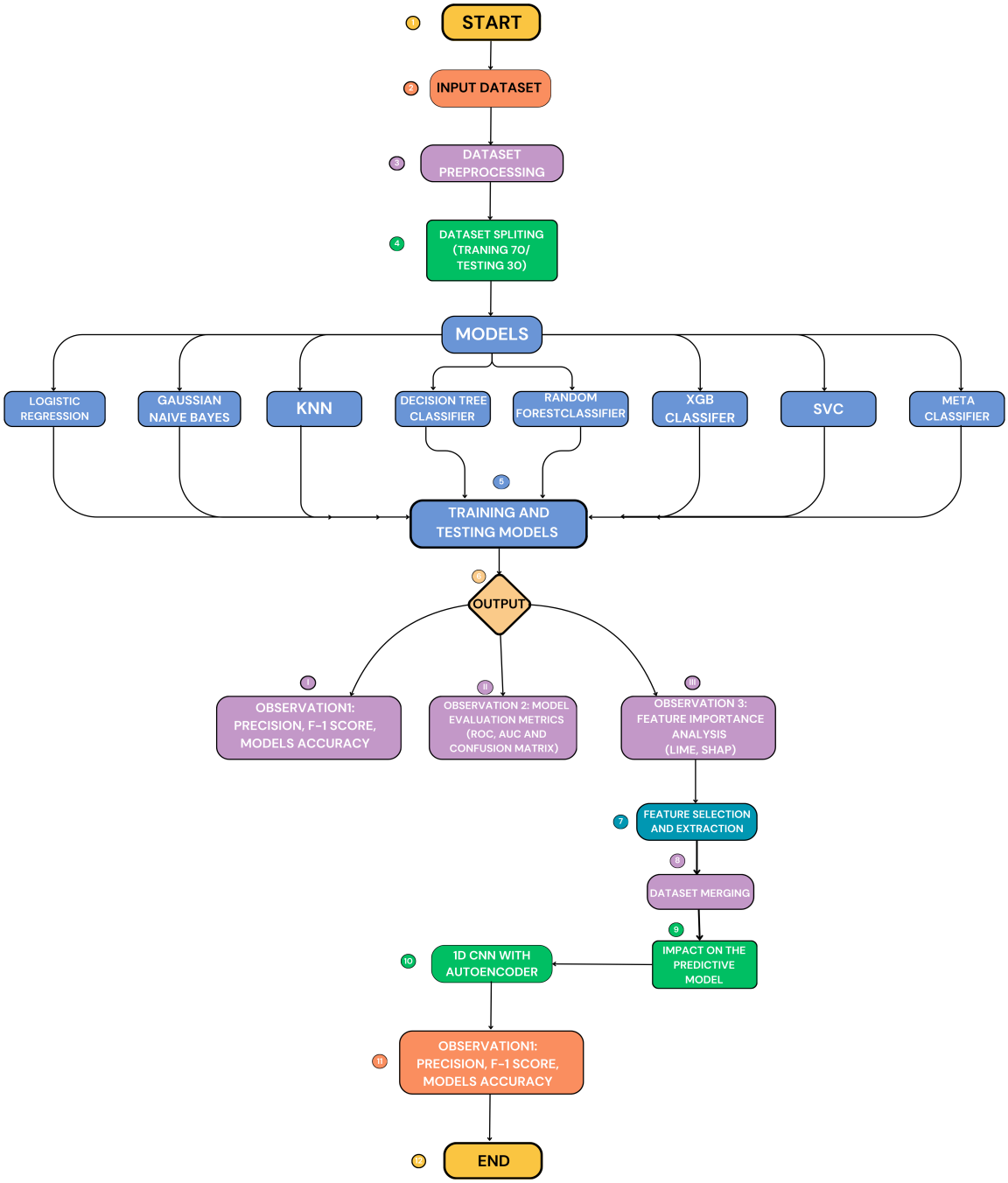



Figure 3.1: Workflow

## 3.2 Model Architecture

In this section, various predictive models will be addressed depending on the research objective. First, some established machine learning algorithms are presented, including K-Nearest Neighbors(KNN), Random Forest(RF), Decision Tree Classifier(DT), XGBoost Classifier(XGB), Support Vector Classifier(SVC), Gaussian Naive Bayes, Logistic Regression, and Meta Classifier. Furthermore, complex architecture including 1D Convolutional Neural Networks(CNN) with autoencoder will also be illustrated for capturing complex patterns and potentially boosting predictive per-

formance. The subsequent subsections discuss the respective architecture and implementation of each model.

### 3.2.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is one of the basic algorithms of machine learning particularly useful for classification and regression. KNN is a non-parametric and instance-based learning method. It makes predictions based on the k-nearest points in the feature space as shown in figure 3.2. In classification, class labels are determined based on majority voting, while in regression problems, class labels are determined based on averaging. Distance metrics like Euclidean distance, Manhattan distance, etc. are used to measure similarity. KNN's nonparametric nature makes it very flexible concerning data distributions since it has very little or no explicit training phase. However, it becomes a little computationally intensive during inference, particularly for large datasets.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

where d(x,y) represents the Euclidean distance between two feature vectors x and y.

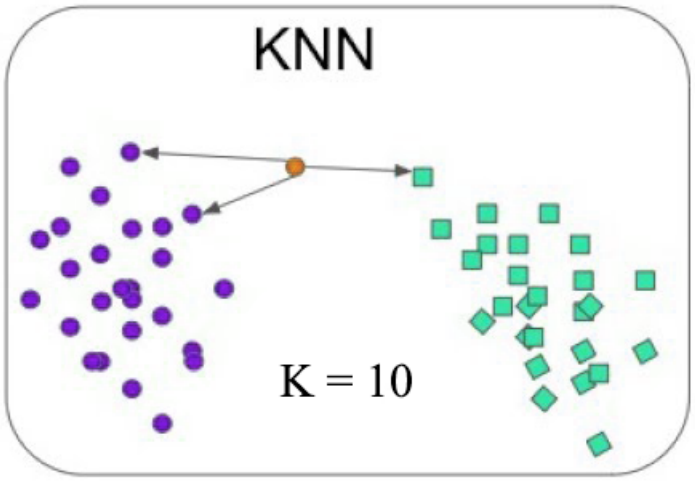



Figure 3.2: K-Nearest Neighbors (KNN) [2]

### 3.2.2 Random Forest Classifier

Random Forest is one collective learning strategy that improves on the classification and regression tasks using an ensemble of decision trees. This method works by constructing numerous decision trees during the training phase and assigning the classes based on either the majority vote from individual trees (for classification) or the average prediction (for regression). With the randomization provided through both the feature selection process and bootstrapped training datasets, it effectively

reduces overfitting while improving generalization. Its approach of introducing bootstrapping and feature randomization allows the Random Forest to be very robust and efficient in handling high dimensionality datasets as shown in figure 3.3.

$$\text{Generalization Error} = \rho(1 - s^2)/s^2 \tag{2}$$

where $\rho$ is the average correlation between pairs of trees, and $s$ is the strength of individual trees.

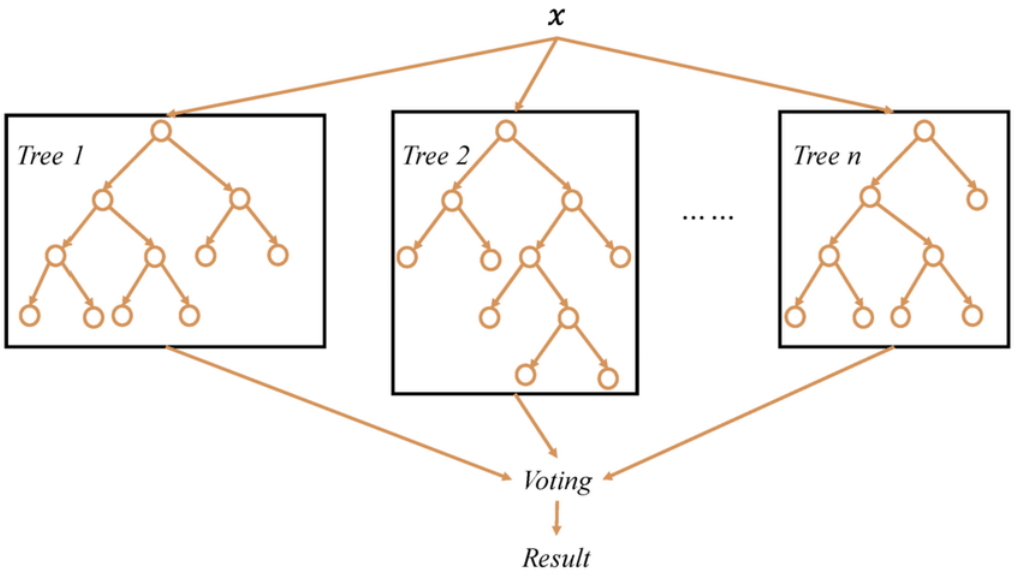



Figure 3.3: Random Forest Classifier [4]

### 3.2.3 Decision Tree Classifier

The Decision Tree Classifier is a rule-based supervised learning algorithm that continuously splits the dataset into subsets according to feature conditions to produce a tree structure. In figure 3.4, the model makes splits based on approaches such as Gini impurity or information gain, seeking to maximize homogeneity within nodes. Decision trees are regarded as easy-to-interpret and computationally efficient tools for smaller datasets. They would have a tendency to overfit unless some pruning techniques are employed.

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \tag{3}$$

where H(S) is entropy before the split, and $H(S_v)$ is entropy after the split.

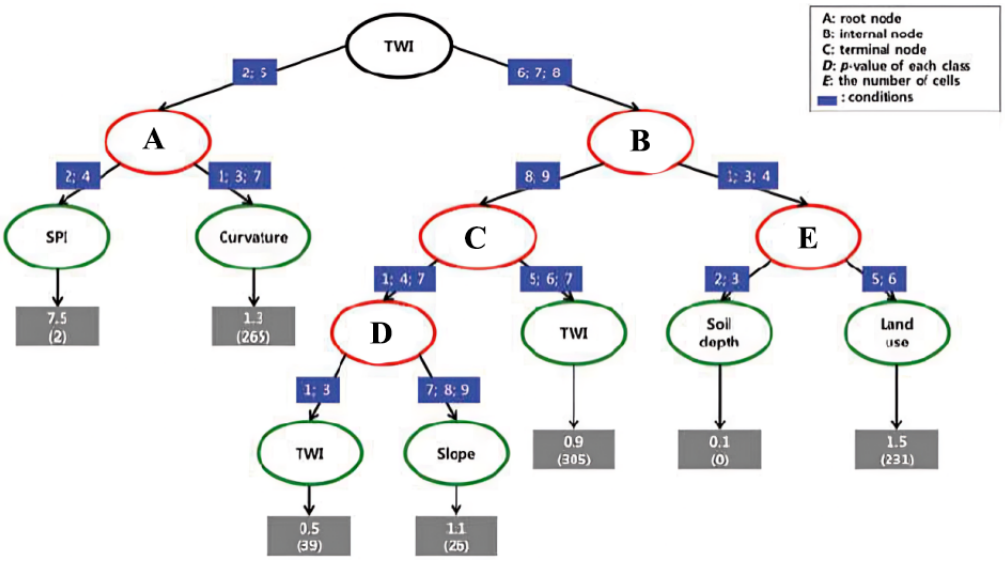



Figure 3.4: Decision Tree Classifier [6]

### 3.2.4 XGBoost Classifier

XGBoost Classifier or Extreme Gradient Boosting is darling of the gradient boosting libraries, boasting superior speed and accuracy. Accoring to figure 3.5, the key innovations include an enhanced booster algorithm exploiting regularization, an approximate solution for growing with the weight of the quantiles, and limited multithreading to help speed up the computation.

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

where $l(y_i, \hat{y}_i)$ is the loss function, and $\Omega(f_k)$ is the regularization term.

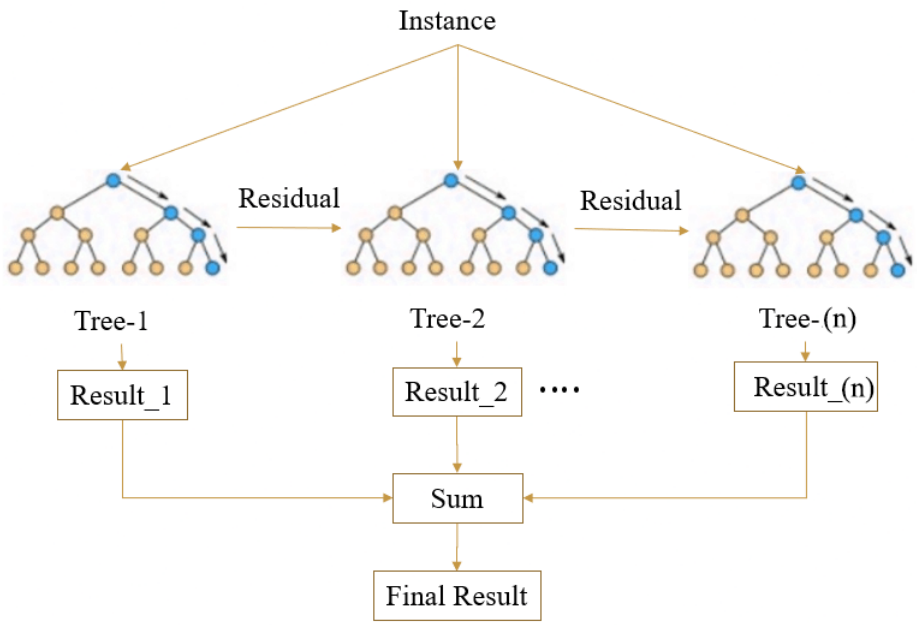



Figure 3.5: XGBoost Classifier [8]

### 3.2.5 Support Vector Classifier (SVC)

As visulaized in figure 3.6, Support Vector Classifier (SVC) is a discriminative classifier by nature. Hence, it seeks an optimal hyperplane such that margin between the classes is maximized. In addition to this, the method also has in its possession the kernel function to deal with the non-linearity, thus allowing for linear, polynomial, and radial basis function(RBF).

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i \tag{5}$$

where w is the weight vector, and b is the bias-term, $y_i$ are the class labels, and $x_i$ are the feature vectors.

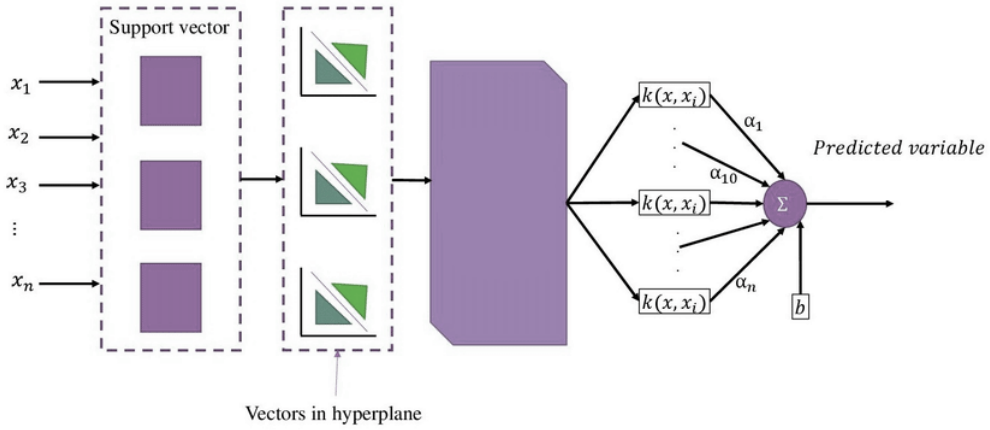



Figure 3.6: Support Vector Classifier (SVC) [3]

### 3.2.6 Gaussian Naive Bayes

Naive Bayes classifiers are probabilistic classifiers based on Bayes' theorem, which assumes the independence of the features. In figure 3.7, it is considered by many to be extremely efficient in text classification, spam filtering, and sentiment analysis due to its computational simplicity and robustness.

$$P(A|\mathbf{X}) = \frac{P(\mathbf{X}|A)P(A)}{P(\mathbf{X})} \tag{6}$$

where Xi represents individual features.

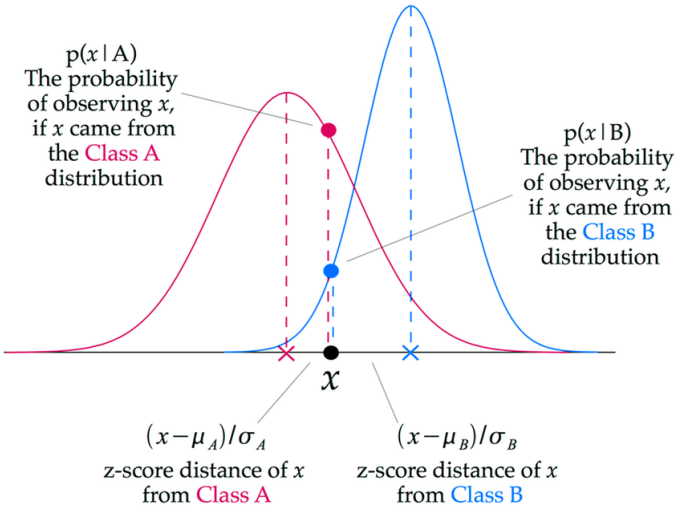



Figure 3.7: Gaussian Naive Bayes [17]

### 3.2.7 Logistic Regression

Logistic regression is a statistical model intended for binary classification or multicategory classification problems. According to figure 3.8, it squashes the output into the (0,1) range, with a natural interpretation as probabilities passes through the sigmoid function.

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{7}$$

where $\beta_0$ is the intercept, and $\beta_1$ is the coefficient.

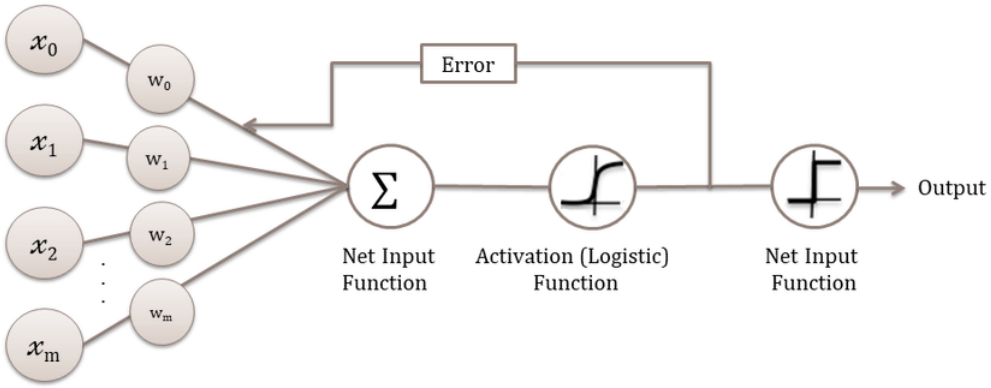



Figure 3.8: Logistic Regression [1]

### 3.2.8 Meta Classifier

A Meta Classifier is an ensemble method that combines multiple fitted base models to improve performance. It uses stacking, boosting, and bagging as common approaches to meta-classification as shown in figure 3.9.

$$\text{Meta\_Consensus\_Output} = \sum_{i=1}^{n} w_i \cdot C_i(x) \tag{8}$$

where:

- n is the number of base classifiers,
- w_i is the weight assigned to the i-th classifier,
- C_i(x) is the output of the i-th classifier for input x.

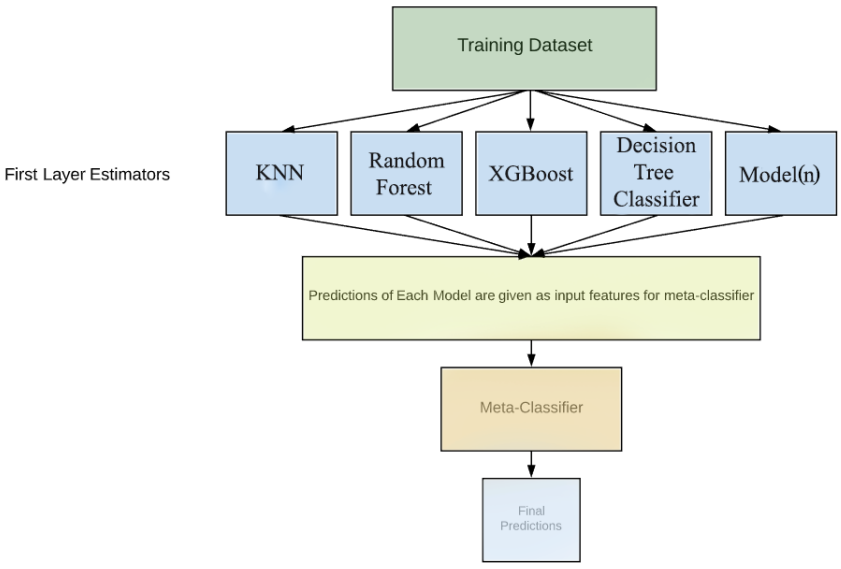



Figure 3.9: Meta Classifier [5]

### 3.2.9 1D Convolutional Neural Network(CNN) with Autoencoder

Using a one-dimensional CNN in a hybrid architecture that includes an autoencoder for feature extraction and classification tasks. The autoencoder is made up of two parts, an encoder which compresses the input data into a much lower-dimensional latent representation, and a decoder which reconstructs the original input from the lower-dimensional representation. The encoder operates on dense layers with ReLU activation functions to perform dimensionality reduction, whereas the decoder uses symmetric layers to reconstruct the input. Training the autoencoder focuses on minimizing the reconstruction error, usually taken care of by the MSE loss function [11]: After training, a meaningful representation from the encoder was reshaped and used to feed the data to a CNN classifier. The classifier consists of convolutional layers intending to capture spatial patterns in the data, alongside batch normalization and dropout layers to improve generalization and avoid overfitting. Finally, the feature maps are flattened and are put through fully connected layers that apply the sigmoid activation function for binary classification tasks. The training of the CNN classifier is done using the BCE loss function [15]. This dual model is efficient in feature learning and hence enhances the classification performance, especially in the case of sequences.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{9.1}$$

where $x_i$ represents the original input, $\hat{x}_i$ denotes the reconstructed input, and $n$ is the number of data points.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{9.2}$$

In this equation, $y_i$ denotes the true binary label, $\hat{y}_i$ is the predicted probability of the positive class, and $n$ is the total number of samples.

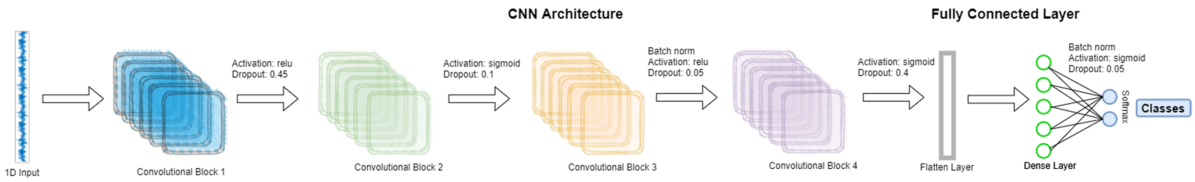



Figure 3.10: 1D Convolutional Neural Network(CNN) with Autoencoder [16]

# Chapter 4

# Machine Learning Techniques for Predictive Modeling

This chapter gives an overview of the entire modeling process, starting from development and evaluation of predictive modeling tools. This chapter delves into many critical aspects, such as dataset characteristics and pre-processing, followed by data splitting and device classification. It outlines specific training parameters that were adopted for different models and discusses the assessment methodologies that were put into practice to evaluate each predictive model performance. In addition, this chapter includes in-depth investigations using accuracy tables, confusion matrices, ROC and AUC curves, and precision, recall, and f1-score, for a wealth of insight into the modeling techniques and their efficiencies.

## 4.1 Dataset Characteristics

Our research used three distinct initial datasets for Alopecia Areata prediction purposes. The initial dataset (a) consists of 999 records alongside 14 features and dataset (b) has 7917 records with 14 features and dataset (c) includes 400 records with 12 features. There are very few features which are common in all three datasets. Most of the features are unique and different across the three datasets. It can be said that the nature of the data collected is vastly different in all three datasets. This diversity amplifies the analysis scope and robustness of predictions.

### 4.1.1 Data Pre-Processing

Imputation was applied to deal with missing values during pre-processing of the datasets used for the research, and further, unnecessary columns like 'Id' and 'Date' that were unrelated with Alopecia Areata were discarded. Then came feature engineering, which encodes categorical features into numerical representations in order to suitably prepare the dataset for machine learning models. With the aim to visualize the interrelationships between features, we employed the Pearson's correlation coefficient formula (10) and, thus, carried out correlation analysis to pinpoint those features which had significant correlation with the target feature 'Hair loss'. From 4.1, correlation heat-map analysis guided the selection of important features and detected multicollinearity in feature-to-feature correlation. Treating multicollinearity not only reduced the dimensionality of the research but also improved performance

and avoided overfitting such that only the relevant features were retained for model training.

$$r_{pcc} = \frac{\sum_{j=1}^{k}(x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{k}(x_j - \bar{x})^2 \sum_{j=1}^{k}(y_j - \bar{y})^2}} \tag{10}$$

Each data point $x_j$ and $y_j$ is subtracted from their respective means $\bar{x}$ and $\bar{y}$, where the product of these deviations is summed over all $k$ data points.

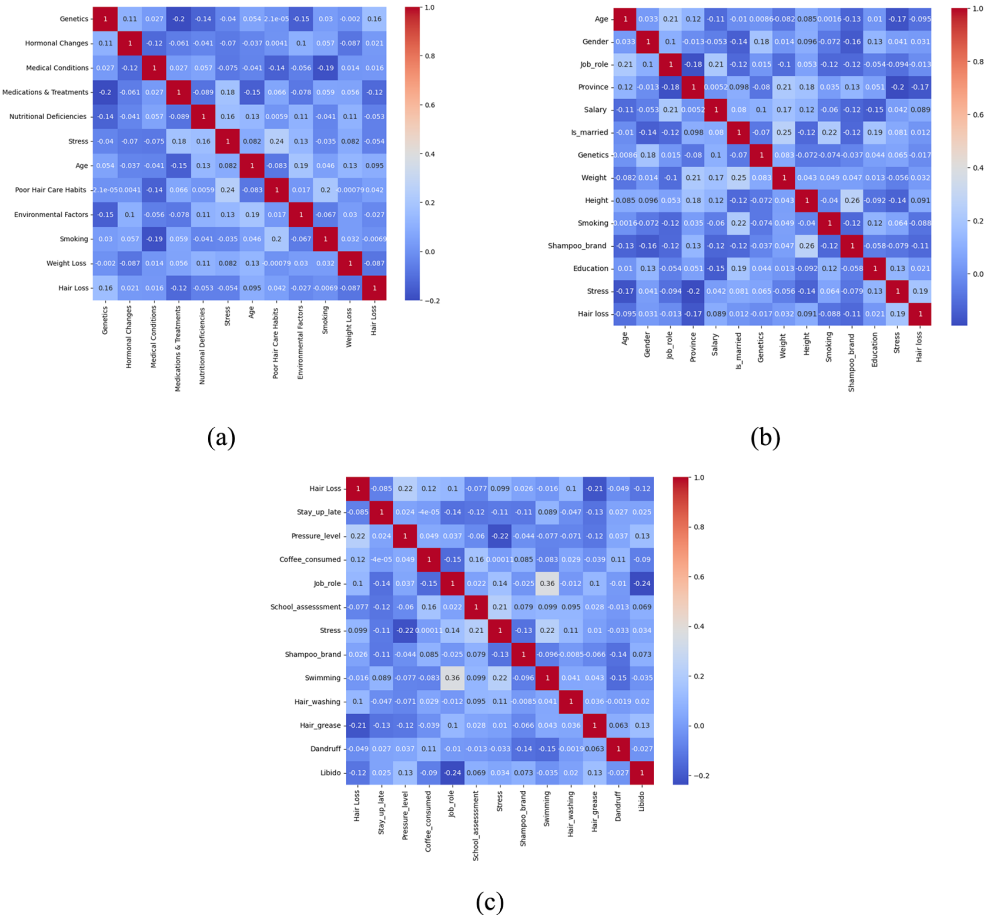



Figure 4.1: Correlation Matrix of the datasets (a), (b) and (c) [14]

### 4.1.2 Splitting Data

During the data splitting stage, the datasets have been divided into 70% of data for training and 30% of data for testing. The application of SMOTE for augmentation was performed only on 70% of the training data pertaining to the three datasets to help reduce class imbalance and better represent the minority class. The remaining 30% test data, however, were not tampered with nor augmented. Thus, preprocessing and augmentation were confined to the training data only in order to avoid possible data leakage while allowing the model to generalize more with unseen data better. After all these processes, we performed scaling, making sure that synthetic samples were properly scaled along with the original data. This preprocessing ensured that the model could learn from balanced and well-scaled data that would, thus, enable better generalization and performance eventually.

### 4.1.3 Device Classification

We have used well 8 well-known machine learning models: Logistic Regression, KNN, Random Forest, DT Classifier, XGB Classifier, SVC, Naive Bayes, and Meta Classifier are used for the given data. All the experiments are performed on multiple machines running Windows 11 Home with AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz and 8GB RAM, Windows 11 Home Single Language 23H2 64-bit operating system, x64-based processor 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz 2.50 GHz RAM 8.00 GB, Windows 10 Home, MSI Intel(R) Core(TM) i5-10500H CPU @ 2.50GHz 2.50 GHz, 16.0 GB RAM. Windows 11 pro with Corei7-14700K with 4080 Super16GB with 64GB DDR5 RAM and 1TB HDD and 1 TB SSD. The performance is assessed based on training accuracy and testing accuracy on the selected datasets. Performance on the examined datasets is quantified by training accuracy and testing accuracy.

### 4.1.4 Training Parameters of Predictive Models

As illustrated in table 4.1, it gives an overview of the various predictive models employed in our reaseach and their parameters for training. Logistic regression runs for a maximum of 500 iterations, while K-Nearest Neighbors uses 10 neighbors. Random Forest Classifier use 300 estimators, while Decision Tree Classifier were initialized with a random state of 42. XGBoost Classifier, Support Vector Classifier, and Gaussian Naive Bayes, involve no specification of iterations or estimators. A stacking ensemble learning approach called 'Meta Classifier' was employed in our research to enhance prediction performance, and it integrates predictive outputs from many base classifiers, such as Support Vector Classifier, Logistic Regression, Random Forest Classifier, K-Nearest Neighbors, Gaussian Naive Bayes, and Decision Tree Classifier. XGBoost was used as a meta-classifier to facilitate learning from the outputs of the base models. All the models use Standard Scaler, letting them enhance the data in a uniform manner, so they are kept mutually preprocessed in all cases.

| Model Name | Iterations/Estimators | Data Enhancement |
|---|---|---|
| Logistic Regression | max_iter=500 | Standard Scaler |
| K-Nearest Neighbors | n_neighbors=10 | Standard Scaler |

| | | |
|---|---|---|
| Random Forest Classifier | n_estimators = 300 | Standard Scaler |
| Decision Tree Classifier | random_state =42 | Standard Scaler |
| XGBoost Classifier | N/A | Standard Scaler |
| Support Vector Classifier (SVC) | N/A | Standard Scaler |
| Gaussian Naive Bayes | N/A | Standard Scaler |

Table 4.1: Summary of Training Parameters

## 4.2 Assessing Predictive Models Performance

This section contains evaluation of the predictive models including Gaussian Naive Bayes, K-Nearest Neighbour, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, XGBoost Classifier, Support Vector Classifier, and Meta Classifier. For performance measurement, confusion matrices, and ROC curves were generated for each model, where a comparative analysis was carried out to observe the influence of these models on datasets (a), (b), and (c). Additionally, the accuracy, precision, recall, and f1-score for all of the predictive models have been calculated to analyze their effectiveness towards classification through comparative study across the datasets.

### 4.2.1 Accuracy Table Summary of Each Predictive Model

The accuracy table 4.2 of various predictive models across the three distinct datasets (a), (b), and (c) highlights the performance variations. Logistic Regression achieved accuracies of 55.00% on dataset (a), 74.87% on dataset (b), and 79.17% on dataset (c). Gaussian Naive Bayes yielded 53.00%, 66.04%, and 70.83% respectively. K-Nearest Neighbors and Decision Tree showed moderate performance, with KNN scoring 48.33%, 48.11%, and 27.50%, while Decision Tree scored 48.00%, 47.35%, and 32.50%. Random Forest Classifier recorded the lowest accuracies, at 45.33%, 45.75%, and 30.83%. XGBoost Classifier also showed varying results, with 49.33% on dataset (a), 47.43% on dataset (b), and 36.67% on dataset (c). Support Vector Classifier performed well, achieving 54.00%, 75.00%, and 78.33%. Lastly, Meta Classifier consistently outperformed other models, with accuracies of 61.00%, 75.55%, and 85.00%, indicating its superior generalization across the datasets.

| **Model Name** | **dataset(a)** Percentage | **dataset(b)** Percentage | **dataset(c)** Percentage |
|---|---|---|---|
| K-Nearest Neighbors | 48.33% | 48.11% | 27.50% |
| Random Forest Classifier | 45.33% | 45.75% | 30.83% |
| Decision Tree Classifier | 48.00% | 47.35% | 32.50% |
| XGBoost Classifier | 49.33% | 47.43% | 36.67% |

| **Model Name** | **dataset(a)** Percentage | **dataset(b)** Percentage | **dataset(c)** Percentage |
|---|---|---|---|
| Support Vector Classifier | 54.00% | 75.00% | 78.33% |
| Gaussian Naive Bayes | 53.00% | 66.04% | 70.83% |
| Logistic Regression | 55.00% | 74.87% | 79.17% |
| Meta Classifier | 61.00% | 75.55% | 85.00% |

Table 4.2: Summary of Model Accuracy

### 4.2.2 Confusion Matrix of Each Predictive Model

**K-Nearest Neighbors:** Looking at Figure 4.2, the confusion matrices for k-Nearest Neighbors (KNN) algorithm shows the various classification problems encountered in datasets (a), (b), and (c). Dataset (a) has a well-distributed pattern of misclassification as the false positive and false negative figures were found to be significant. Dataset (b), due to its larger sample size, experiences misclassifications highlighting moderate prediction imbalance. In contrast, dataset (c), recorded as the smallest in sample size, presented high skewness in classification as it had very few true negatives and a considerable amount of false negatives. Overall, dataset (b) has the most considerable challenge of high misclassifications, whereas dataset (c) has very high challenges concerning imbalance.

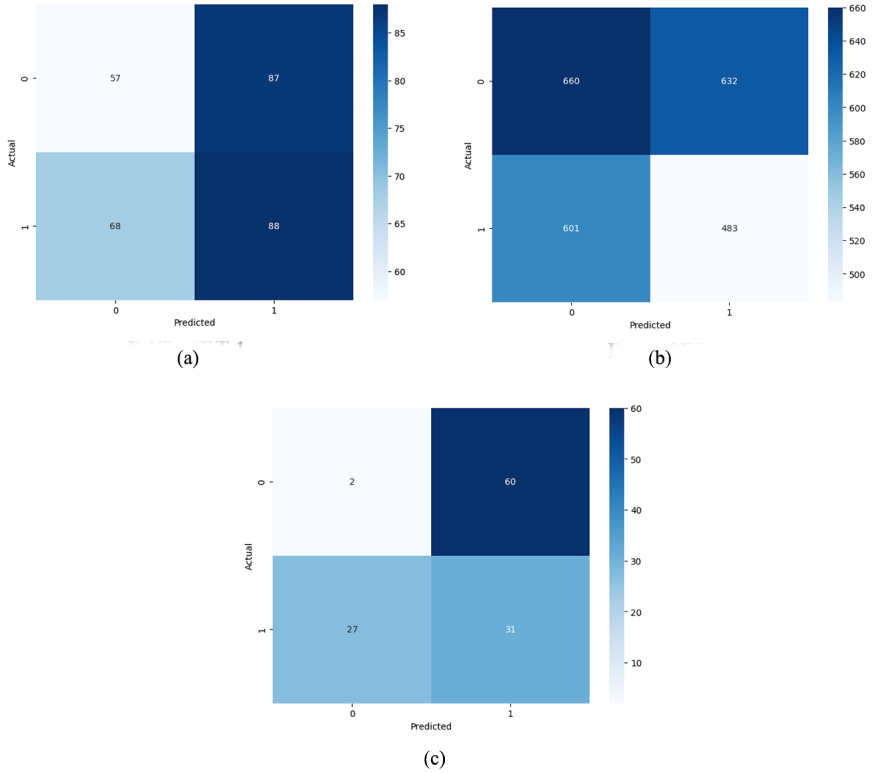



Figure 4.2: Dataset (a) Confusion Matrix of K-Nearest Neighbors, Dataset (b) Confusion Matrix of K-Nearest Neighbors and Dataset (c) Confusion Matrix of K-Nearest Neighbors

**Random Forest Classifier:** According to Figure 4.3, the confusion matrices of Random Forest over datasets (a), (b), and (c) show different performances. Dataset (a) has a problem of class imbalance that obtains high rates of false positives and false negatives. Balanced classification for dataset (b) includes misclassifications in both classes. In particular, dataset (c) shows a predilection toward predicting one class, and hence very few true negatives. This inconsistency in performance is more pronounced under Random Forest, with dataset (b) showing the highest balances achievable, while datasets (a) and (c) encountered difficulties in both classes.

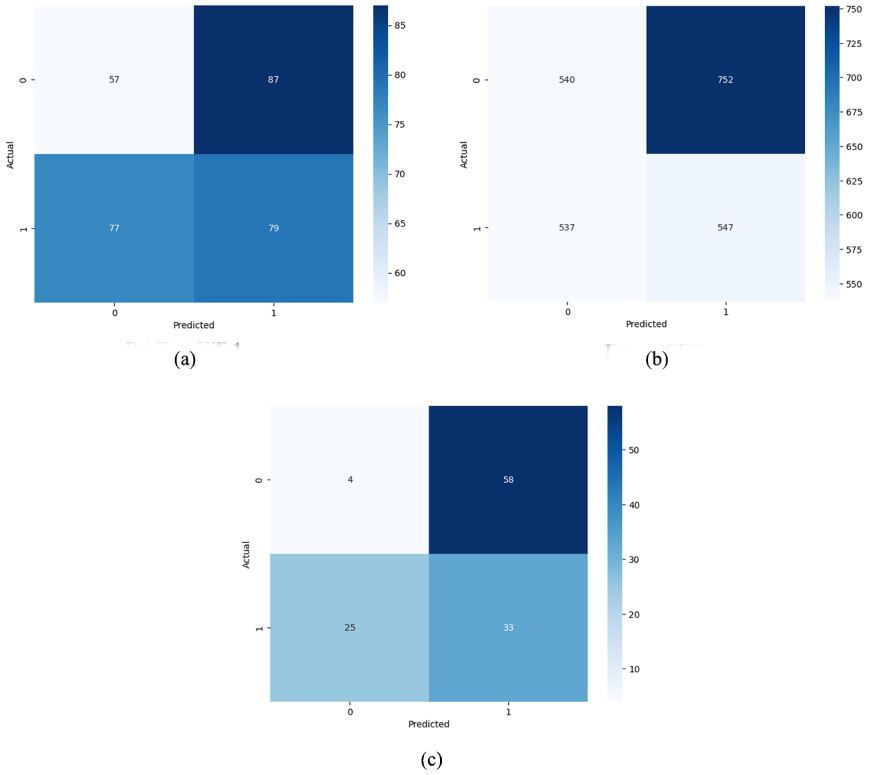



Figure 4.3: Dataset (a) Confusion Matrix of Random Forest Classifier, Dataset (b) Confusion Matrix of Random Forest Classifier and Dataset (c) Confusion Matrix of Random Forest Classifier

**Decision Tree Classifier:** From Figure 4.4, it is evident that within each datasets (a), (b), and (c), the performance exhibited by the Decision Tree Classifier varies significantly.Dataset (a) has somewhat good balance between true positives and true negatives, but it features a lot of false negatives. On the contrary, dataset (b) seems to have the highest total counts in all categories, notably because it features glaring false positives and enormous numbers of false negatives. The smallest data among the three sets is dataset (c), which shows a lot more false positives compared to the figure of true negatives, with moderate figures of false negatives and positives. These features highlight how the characteristics of different datasets affect the model performances.

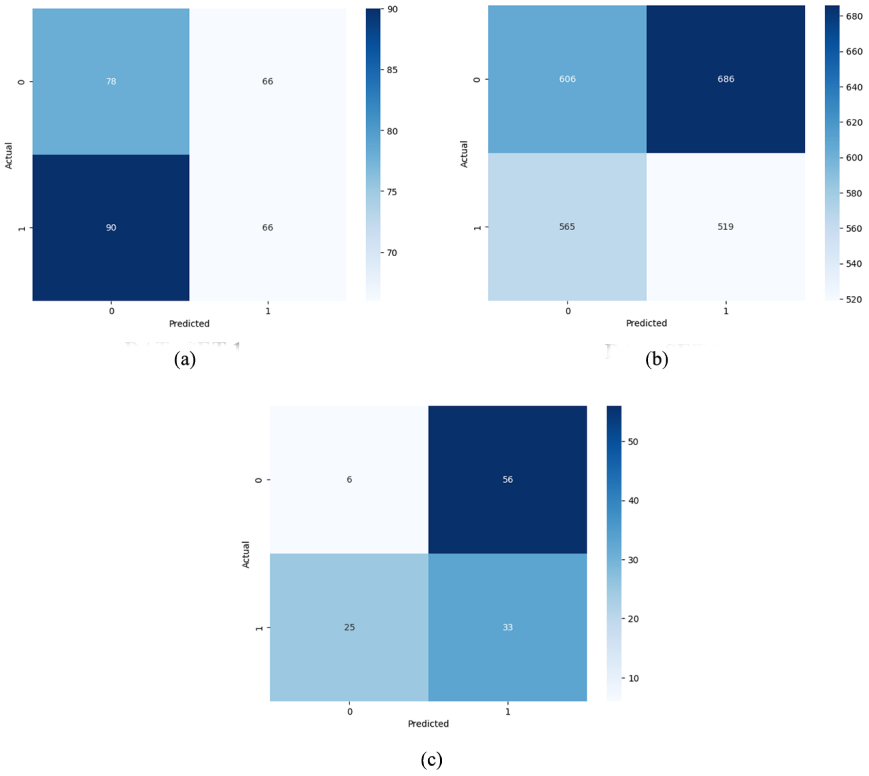



Figure 4.4: Dataset (a) Confusion Matrix of Decision Tree Classifier, Dataset (b) Confusion Matrix of Decision Tree Classifier and Dataset (c) Confusion Matrix of Decision Tree Classifier

**XGBoost Classifier:** Referring to Figure 4.5, the confusion matrices of XGBoost indicates the performance obtained across the datasets (a), (b), and (c). The higher false positive rate, especially for dataset (a), highlights the poor ability of class separability, followed by a high rate of false-negative classification.The similar high misclassifications of both classes in dataset (b) suggest poor classification. On the other hand, dataset (c) performed excellently with near perfection in classes with less misclassification and high predictive accuracy. XGBoost actually worked on dataset (c) ,but poor classification problems were confronted with great difficulty in datasets (a) and (b).

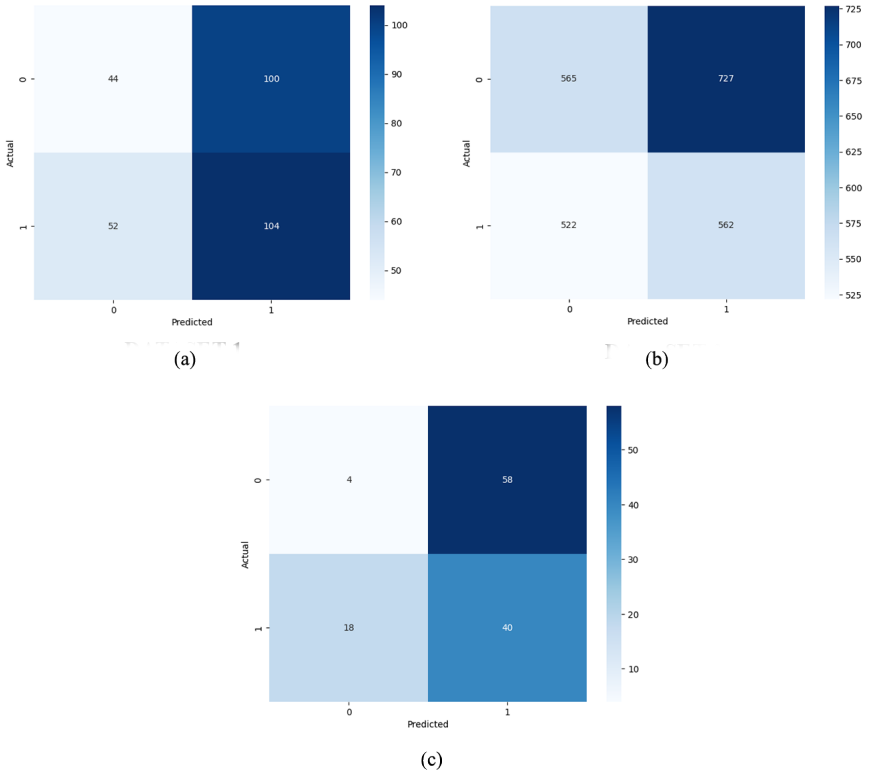



Figure 4.5: Dataset (a) Confusion Matrix of XGBoost Classifier, Dataset (b) Confusion Matrix of XGBoost Classifier and Dataset (c) Confusion Matrix of XGBoost Classifier

**Support Vector Classifier:** As shown in figure 4.6, the confusion matrices for Support Vector Classification across datasets (a), (b), and (c) vary in performance. For dataset (a), misclassification numbers are quite high on both classes. Dataset (b) is performing much better, there is more or less balanced classification and lower false positives or false negatives. Dataset (c) gives more accurate results with very few false positives and clear differentiation between both classes. Generally, SVC works the best for dataset (c) while dataset (a) suffers from heavy classification problems.

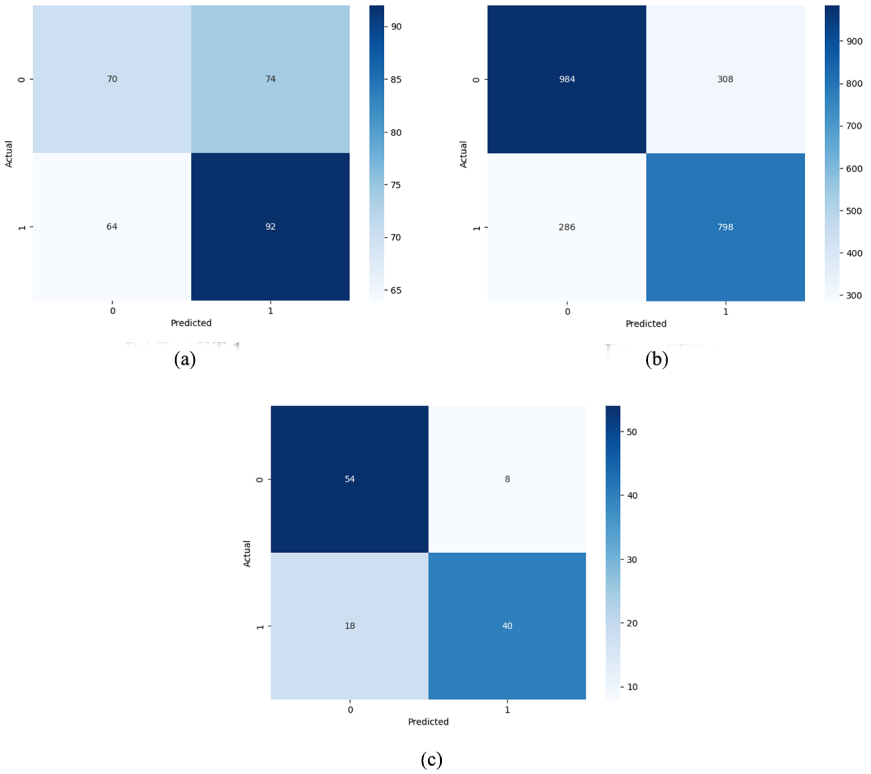



Figure 4.6: Dataset (a) Confusion Matrix of Support Vector Classifier, Dataset (b) Confusion Matrix of Support Vector Classifier and Dataset (c) Confusion Matrix of Support Vector Classifier

**Gaussian Naive Bayes:** From the results in Figure 4.7, the confusion matrices pertaining to Naive Bayes with respect to datasets (a), (b), and (c) show huge dissimilarities in its performance. For example, in dataset (a), the false positives are higher but false negatives are relatively lower indicating that it is more biased towards overpredicting the positive class. Dataset (b) counts a balance between the classes with much higher figures for false negatives, making it appear not very effective to deliver a good positive class prediction. The smaller sample size of dataset (c) illustrates a fairly and moderately balanced performance with false positives and false negatives. Comparisons such as this show how temporarily an effectiveness of the Naive Bayes may result in differences among datasets, thereby justifying model evaluations focusing on particular settings.

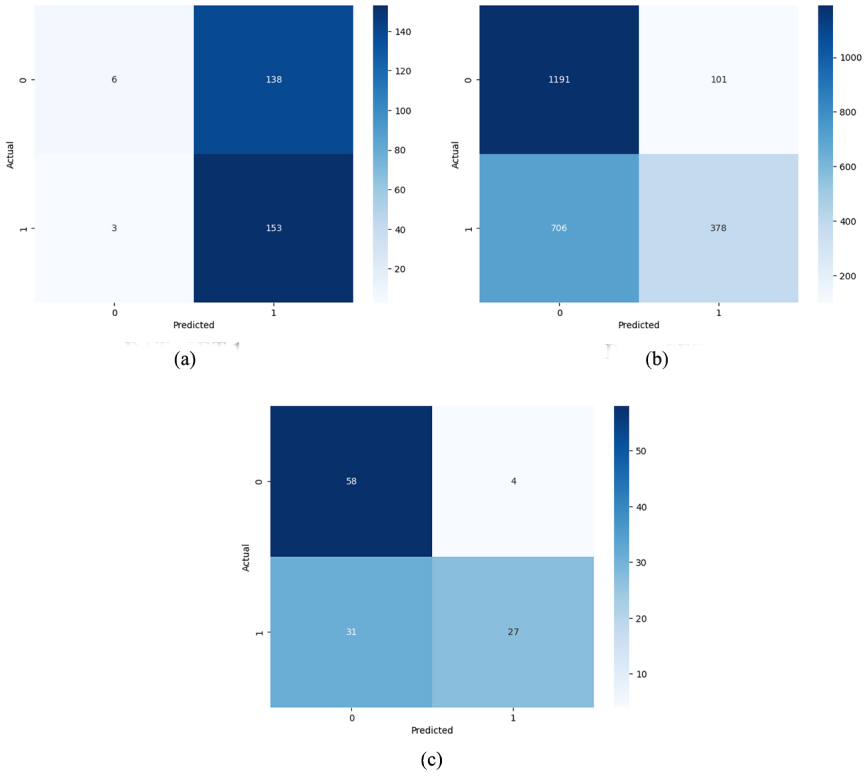



Figure 4.7: Dataset (a) Confusion Matrix of Gaussian Naive Bayes, Dataset (b) Confusion Matrix of Gaussian Naive Bayes and Dataset (c) Confusion Matrix of Gaussian Naive Bayes

**Logistic Regression:** As illustrated in Figure 4.8, the confusion matrices from logistic regression results across the three datasets for (a), (b), and (c) also show differences in the classification performances employed here. Dataset (a) seems to portray a relatively even misclassification pattern, recording a sizeable number of false positives and a false negative observation. Dataset (b) has a larger sample size, displays richer results compared to the other datasets with much more correct than wrong classifications of which indicate stronger predictive capabilities. Dataset (c) has recorded a significantly enhanced level of accuracy, as it scored very high on true positives and true negatives while keeping false classifications at a bare minimum. All in all, logistic regression seems to be performing best on dataset (c) while dataset (a) is experiencing quite a few misclassification problems.

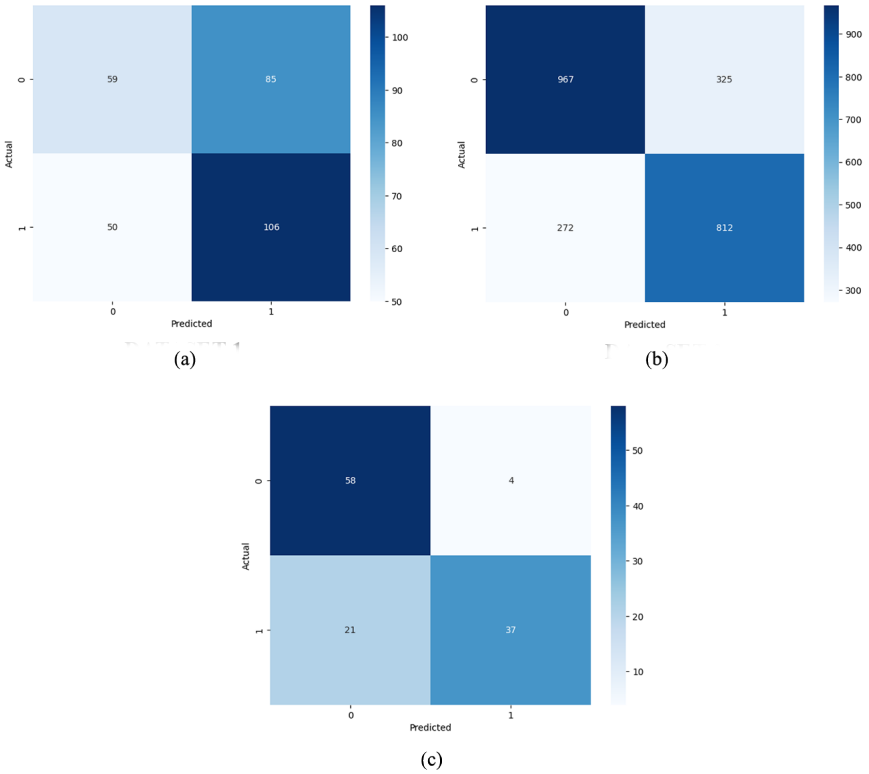



Figure 4.8: Dataset (a) Confusion Matrix of Logistic Regression, Dataset (b) Confusion Matrix of Logistic Regression and Dataset (c) Confusion Matrix of Logistic Regression

**Meta Classifier:** Observing Figure 4.9, the confusion matrices of the meta-classifier across 3 datasets (a),(b), and (c) show marked improvements in classification effectiveness. With dataset (a), some balance in misclassification has been achieved, but the issue of false positives remains significant. Dataset (b) shows excellent results with its larger sample size, which is able to yield an increased amount of correct classifications against diminished errors when compared to the previous models. The accuracy of dataset (c) is maximized, with trivial amounts of misclassification, as evidenced by high values of true positives and true negatives. As can be seen from this comparison, the various performances of the meta classifier can be observed in different datasets, where dataset (c) yielded the most accurate results.

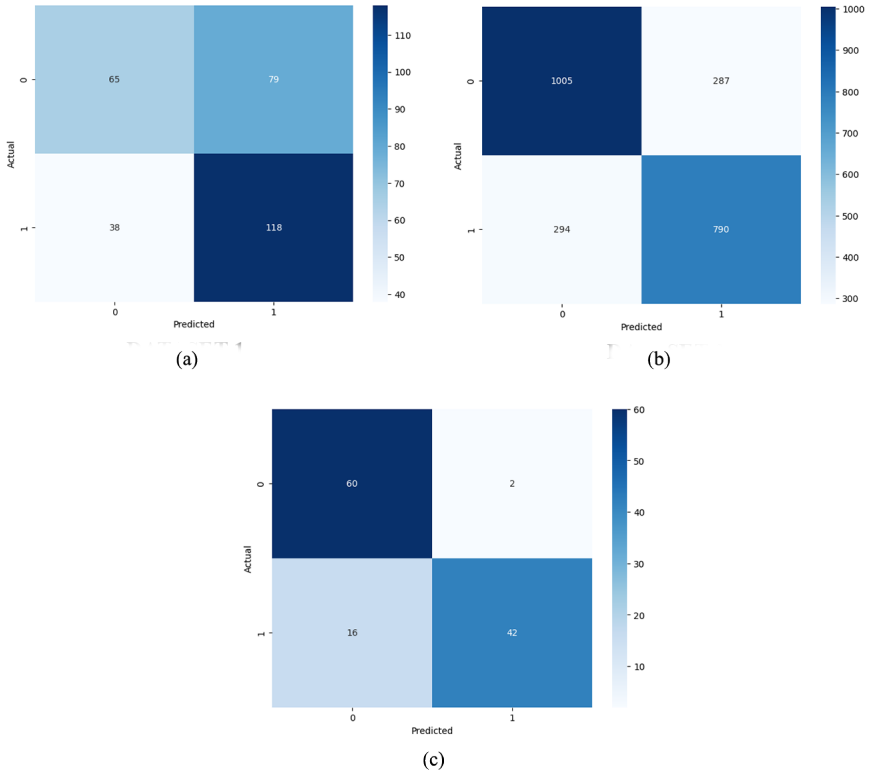



Figure 4.9: Dataset (a) Confusion Matrix of Meta Classifier, Dataset (b) Confusion Matrix of Meta Classifier and Dataset (c) Confusion Matrix of Meta Classifier

### 4.2.3 ROC and AUC of Each Predictive Model

**K-Nearest Neighbors:** As seen in figure 4.10, the K-Nearest Neighbors performs differently in these three databases. In the first dataset (a), the AUC value is 0.5510 for "Hair Fall" and 0.4490 for "No Hair Fall", which is approximately random performance. Dataset (b) has AUC values of 0.7853 (representing "Hair Fall") and 0.2147 (for "No Hair Fall"). In dataset (c), the "hair fall" AUC is 0.8133 and that for "No Hair Fall" is 0.1867. So, in figure 4.10, the KNN classifier does well for "hair fall" in datasets (b) and (c); however, the performance for "No Hair Fall" remains dismally low across datasets.

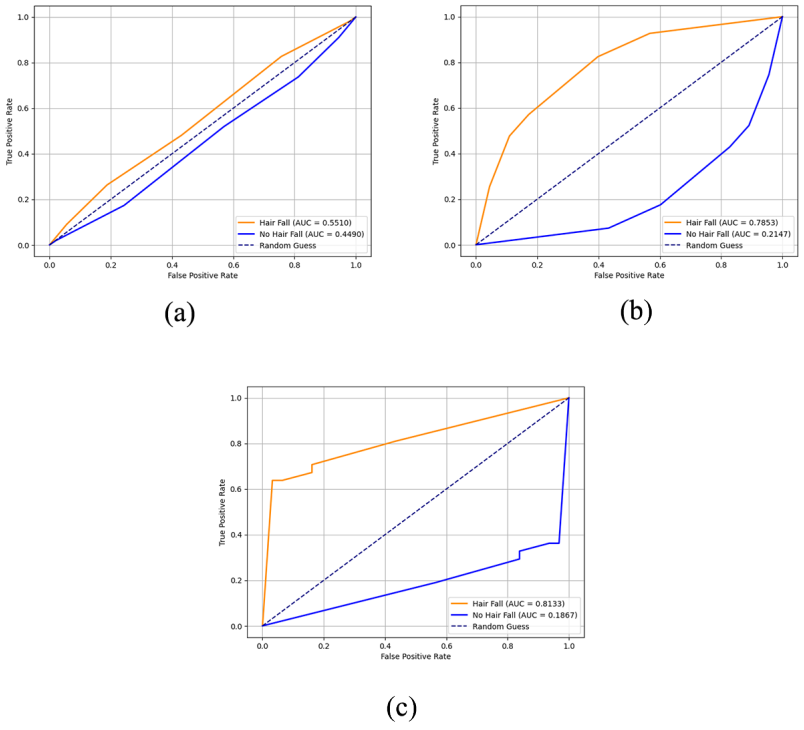



Figure 4.10: Dataset (a) ROC_AUC Curve for K-Nearest Neighbors, Dataset (b) ROC_AUC Curve for K-Nearest Neighbors and Dataset (c) ROC_AUC Curve for K-Nearest Neighbors

**Random Forest Classifier:** In figure 4.11, different trends are seen in the performance of the Random Forest classifier on the three datasets. For dataset (a), the values of AUC are 0.5331 for "Hair Fall" and 0.4669 for "No Hair Fall", indicating performance near that of a random classifier. For dataset (b), the classifier performs very well for the class "Hair Fall" with an AUC of 0.8219, whereas for class "No Hair Fall" its performance is very poor with an AUC of 0.1781. This trend is evident as well in dataset (c), where the AUC for "Hair Fall" is 0.8254, while it is 0.1746 for "No Hair Fall". Thus, from figure 4.11, it can be said that for the "Hair Fall" class, the classifier works well for datasets (b) and (c), but it works poorly for the "No Hair Fall" class across all datasets.

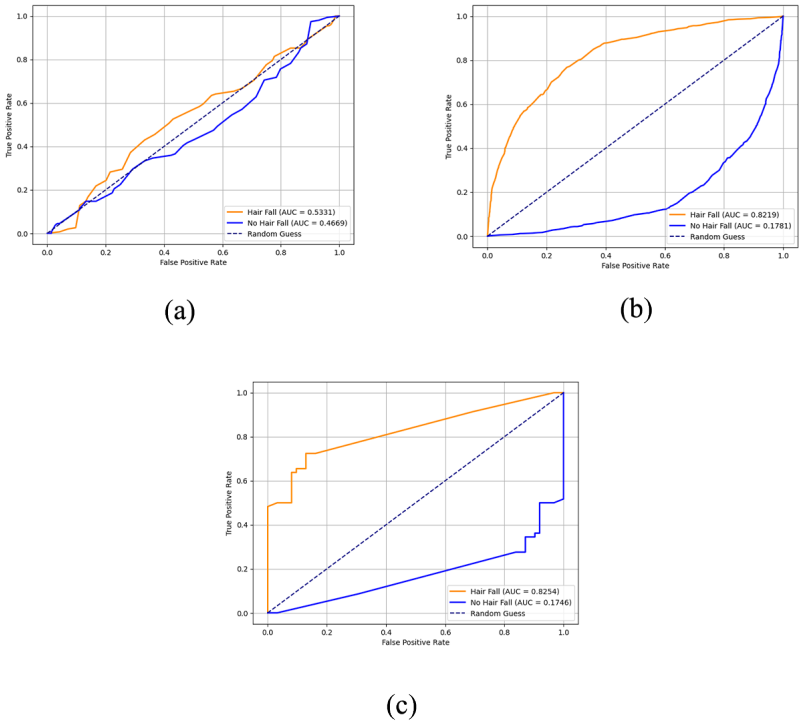



Figure 4.11: Dataset (a) ROC_AUC Curve for Random Forest Classifier, Dataset (b) ROC_AUC Curve for Random Forest Classifier and Dataset (c) ROC_AUC Curve for Random Forest Classifier

**Decision Tree Classifier:** From figure 4.12, we can see that the graphs compare the performance of a Decision Tree Classifier with three different datasets, (a), (b), (c) with ROC curves and the corresponding AUC values. The classifier had an AUC value of 0.5173 for dataset (a) to "Hair Fall" and 0.4827 to "No Hair Fall" indicating an almost random performance. The classifier performs better in dataset (b) compared to an AUC of 0.8218 with "Hair Fall" but poorly with an AUC of 0.1782 for "No Hair Fall". The same applies for dataset (c), where the classifier has an AUC of 0.8315 for "Hair Fall" but falls extremely low at 0.1685 for "No Hair Fall". So in figure 4.12,the classifier overall does well for the "Hair Fall" class in both datasets (b) and (c), but almost gives random results in dataset (a). The "No Hair Fall" class, however, consistently fails in all datasets, although remarkably exhibits strong results on the "Hair Fall" class in both datasets (b) and (c).

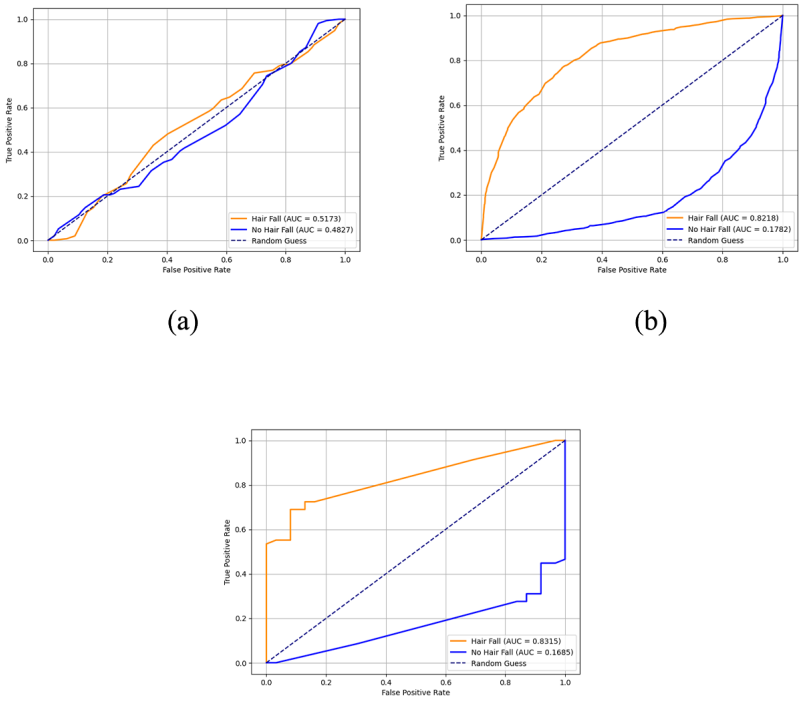



Figure 4.12: Dataset (a) ROC_AUC Curve for Decision Tree Classifier, Dataset (b) ROC_AUC Curve for Decision Tree Classifier and Dataset (c) ROC_AUC Curve for Decision Tree Classifier

**XGBoost Classifier:** From figure 4.13, we can see that across these three datasets, the behavior of the XGBoost classifier does not remain the same. The dataset (a) contains AUC values of 0.5380 for "Hair Fall" and 0.4620 for "No Hair Fall," suggesting almost random performance. In dataset (b), on the other hand, the classifier indicates a good performance for "Hair Fall" with an AUC of 0.8219 and a terrible performance for "No Hair Fall" with an AUC of 0.1781. Dataset (c) yields similar results, showing AUC scores: Hair Fall, 0.8215 and No Hair Fall, 0.1785. All in all,in figure 4.13, the model does well at predicting Hair Fall for datasets (b) and (c), but No Hair Fall performance is consistently poor across the datasets.

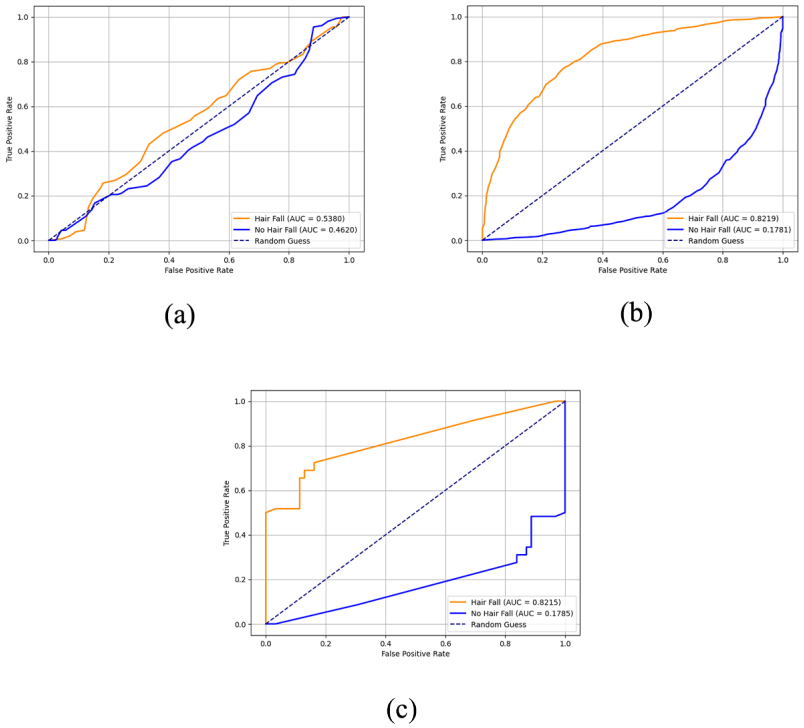



Figure 4.13: Dataset (a) ROC_AUC Curve for XGBoost Classifier, Dataset (b) ROC_AUC Curve for XGBoost Classifier and Dataset (c) ROC_AUC Curve for XGBoost Classifier

**Support Vector Classifier:** In figure 4.14, the graph shows the ROC curves and the AUC values for analysis of an SVC classifier on three datasets labeled as (a), (b), and (c). For dataset (a), it has been found that the classifier has poor performance with AUCs of 0.4447 for "Hair Fall" and 0.5553 for "No Hair Fall." For "Hair Fall," this classifier obtained a higher score of 0.7876 in dataset (b), while in comparison, it had a low score of 0.2124 for "No Hair Fall." Likewise, in dataset (c), the classifier scored AUC for "Hair Fall" up to 0.7981 and for "No Hair Fall" it got down to 0.2019. In figure 4.14, it is fair to say that as compared to 'No Hair Fall', it is better in terms of "Hair Fall" for datasets (b) as well as (c), while still being consistently not-so-good for No Hair Fall in all datasets.

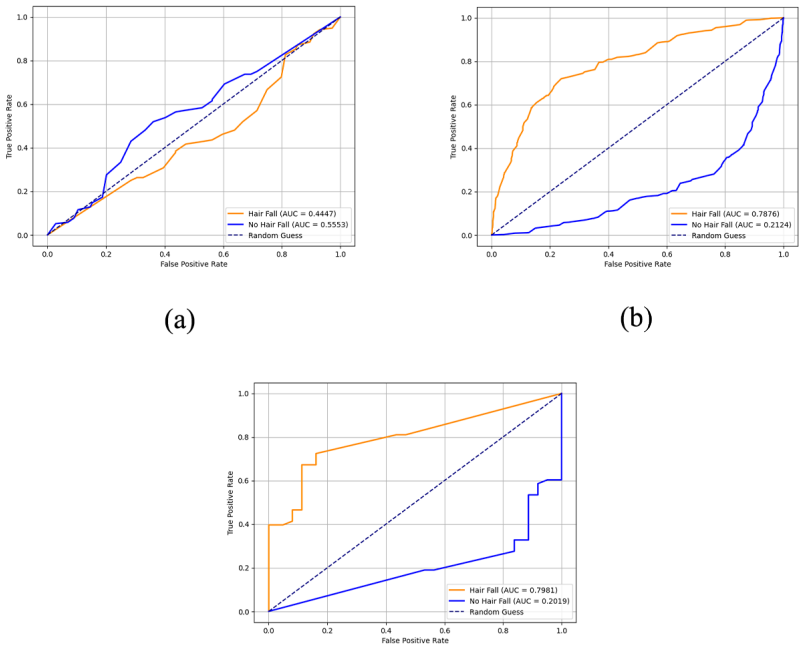



Figure 4.14: Dataset (a) ROC_AUC Curve for Support Vector Classifier, Dataset (b) ROC_AUC Curve for Support Vector Classifier and Dataset (c) ROC_AUC Curve for Support Vector Classifier

**Gaussian Naive Bayes:** Based on figure 4.15, for all three datasets, the performance of the Naive Bayes classifier is as follows: For dataset (a), the AUC values near 0.5 indicate performance close to random for "Hair Fall" (0.5967) and "No Hair Fall" (0.4033). Dataset (b) shows very good performance with AUC scores of 0.8153 for "Hair Fall" and 0.8147 for "No Hair Fall." The AUC value is 0.8115 for "Hair Fall" in dataset (c), while for "No Hair Fall," it is very low with an AUC of 0.1885. Overall, in figure 4.15 ,across both classes, dataset (b) shows the best performance, while dataset (c) does well for "Hair Fall" but very poorly for "No Hair Fall."

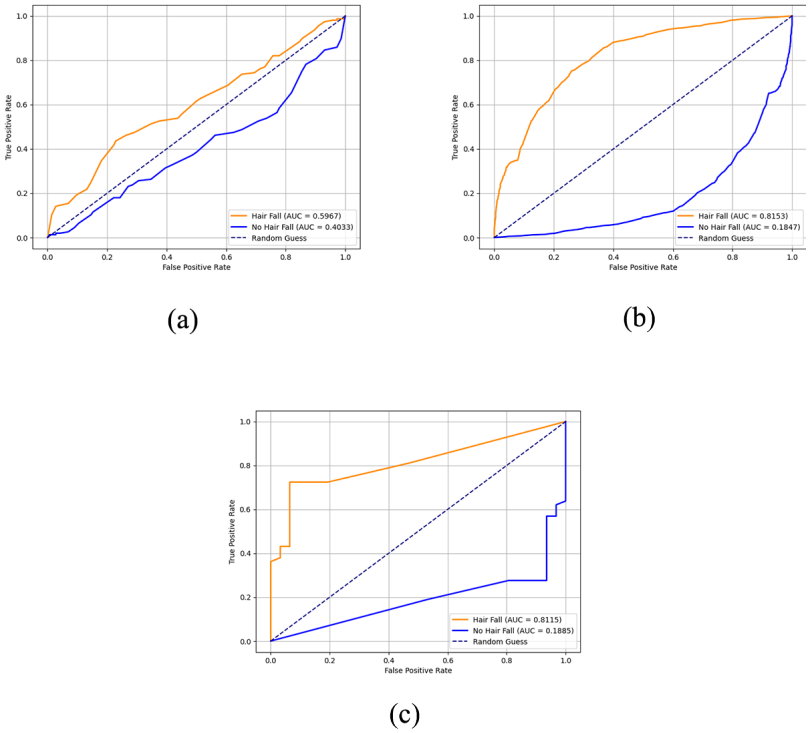



Figure 4.15: Dataset (a) ROC_AUC Curve for Gaussian Naive Bayes, Dataset (b) ROC_AUC Curve for Gaussian Naive Bayes and Dataset (c) ROC_AUC Curve for Gaussian Naive Bayes

**Logistic Regression:** As seen in figure 4.16, the graphs that show the different ROC curve comparisons and even AUC values provide an insight into the performance of the Logistic Regression model on three datasets termed as (a), (b), and (c) respectively. The model that has been run on dataset(a) gives the corresponding AUC values of 0.5870 against "Hair Fall" and 0.4130 in "No Hair Fall", which actually just indicates a little bit better performance than random. As for dataset (b), while the AUC for "Hair Fall" stands at a great 0.8249, the AUC for "No Hair Fall" is low, at only 0.1751. The same way it does for dataset (c), where the AUC for "Hair Fall" is 0.8148, and for "No Hair Fall" it stands at only 0.1852. Overall, from figure 4.16, we can generalize that the Logistic Regression model gave satisfactory results with reference to "Hair Fall" prediction for datasets (b) and (c); its performance for prediction of "No Hair Fall" does not come very high in any of the datasets.

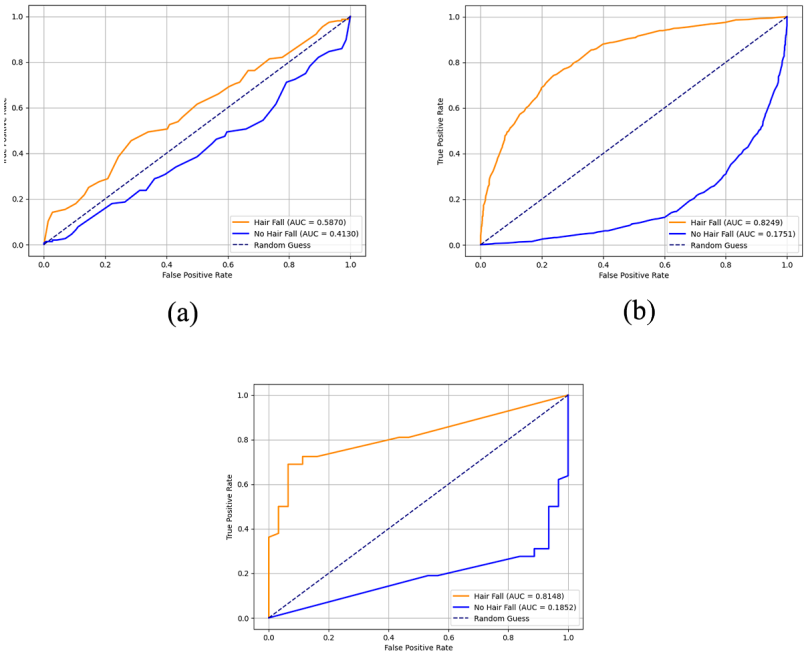



Figure 4.16: Dataset (a) ROC_AUC Curve for Logistic Regression, Dataset (b) ROC_AUC Curve for Logistic Regression and Dataset (c) ROC_AUC Curve for Logistic Regression

**Meta Classifier:** In figure 4.17, the performance of the Meta Classifier on the three datasets was observed as follows: Dataset (a) presents a poor discrimination ability by the classifier, with AUC values of 0.5380 for "Hair Fall" and 0.4620 for "No Hair Fall". On the other hand, datasets (b) and (c) showed very strong and consistent performance with AUC results of around 0.8219 for dataset (b) and 0.8215 for dataset (c)in "Hair Fall" and 0.7181 for dataset (b) and 0.7185 for dataset (c) under the "No Hair Fall". The overall performance in figure 4.17 indicates that while it performs poorly on dataset (a), it did soundly well on datasets (b) and (c).

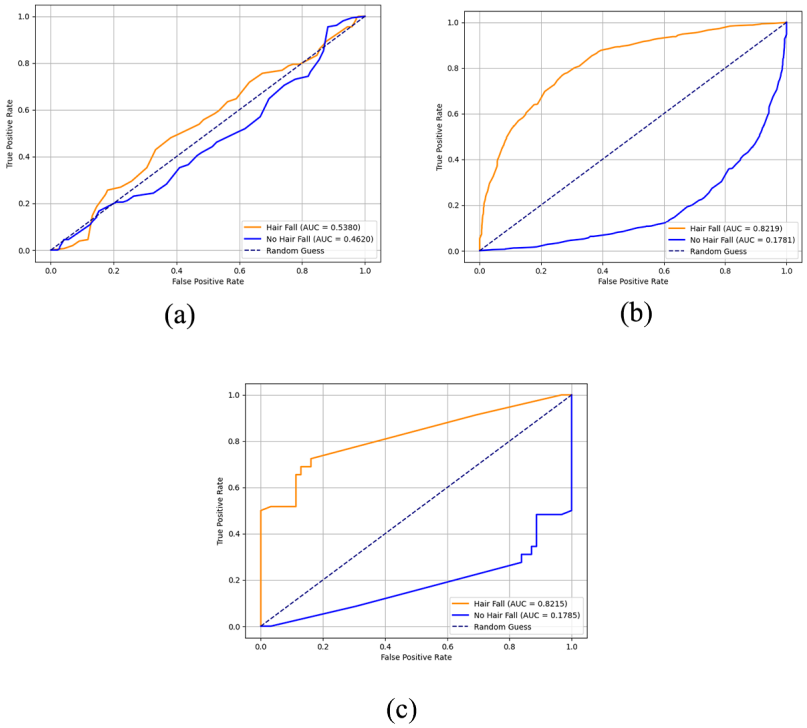



Figure 4.17: Dataset (a) ROC_AUC Curve for Meta Classifier, Dataset (b) ROC_AUC Curve for Meta Classifier and Dataset (c) ROC_AUC Curve for Meta Classifier

### 4.2.4 Precision, Recall and F1-Score of Each Predictive Model

**K-Nearest Neighbors:** From table, 4.3 we can see that, dataset (a): precision (0.5029), recall (0.5641), and f1-score (0.5317) reflect moderate performance, dataset (b): precision (0.4332), recall (0.4456), and f1-score (0.4393) indicate below-average performance and dataset (c): precision (0.3407), recall (0.5345), and f1-score (0.4161) reveal weaker performance.

**Random Forest Classifier:** The table of 4.3 stated that, dataset (a): precision (0.4759), recall (0.5064), and f1-score (0.4907) reflect moderate performance, dataset (b): precision (0.4211), recall (0.5046), and f1-score (0.4591) suggest below-average performance and dataset (c): precision (0.3626), recall (0.5690), and f1-score (0.4430) indicate weaker performance.

**Decision Tree Classifier:** According to table, 4.3 dataset (a): precision (0.5000), recall (0.4231), and f1-score (0.4583) reflect moderate performance, dataset (b): precision (0.4307), recall (0.4788), and f1-score (0.4535) indicate below-average performance and dataset (c): Precision (0.3708), recall (0.5690), and f1-score (0.4490) show weaker performance.

**XGBoost Classifier:** Dataset (a): precision (0.5098), recall (0.6667), and f1-score (0.5778) demonstrate better results compared to other models on this dataset according to the table 4.3. Dataset (b): precision (0.4360), recall (0.5185), and f1-score (0.4737) suggest moderate performance. Moreover, dataset (c): precision (0.4082), recall (0.6897), and f1-score (0.5128) reflect moderate performance.

**Support Vector Classifier:** From table 4.3, we observe that, dataset (a): preci-

sion (0.5542), recall (0.5897), and f1-score (0.5714) indicate relatively better performance, dataset (b): precision (0.7215), recall (0.7362), and f1-score (0.7288) indicate strong performance and dataset (c): precision (0.8333), recall (0.6897), and f1-score (0.7547) reflect strong performance.

**Gaussian Naive Bayes:** Dataset (a): precision (0.5258), recall (0.9808), and f1-score (0.6846) show high recall but lower precision according to the table 4.3. Dataset (b): precision (0.7891), recall (0.3487), and f1-score (0.4837) indicate below-average performance. Dataset (c): precision (0.8710), recall (0.4655), and f1-score (0.6067) reflect moderate performance.

**Logistic Regression:** According to table 4.3, dataset (a): precision (0.5550), recall (0.6795), and f1-score (0.6110) suggest moderate performance, dataset (b): precision (0.7142), recall (0.7491), and f1-score (0.7312) indicate strong performance and dataset (c): precision (0.9024), recall (0.6379), and f1-score (0.7475) reflect strong performance.

**Meta Classifier:** The table 4.3 demonstrated that, dataset (a): precision (0.5990), recall (0.7564), and f1-score (0.6686) suggest moderate to strong performance, dataset (b): precision (0.7865), recall (0.7418), and f1-score (0.7636) indicate strong performance and dataset (c): precision (0.8615), recall (0.8148), and f1-score (0.8376) reflect exceptional performance.

| **Model** | Precision dataset (a) | Recall dataset (a) | F1-score dataset (a) | Precision dataset (b) | Recall dataset (b) | F1-score dataset (b) | Precision dataset (c) | Recall dataset (c) | F1-score dataset (c) |
|---|---|---|---|---|---|---|---|---|---|
| K-Nearest Neighbors | 0.5029 | 0.5641 | 0.5317 | 0.4332 | 0.4456 | 0.4393 | 0.3407 | 0.5345 | 0.4161 |
| Random Forest Classifier | 0.4759 | 0.5064 | 0.4907 | 0.4211 | 0.5046 | 0.4591 | 0.3626 | 0.5690 | 0.4430 |
| Decision Tree Classifier | 0.5000 | 0.4231 | 0.4583 | 0.4307 | 0.4788 | 0.4535 | 0.3708 | 0.5690 | 0.4490 |
| XGBoost Classifier | 0.5098 | 0.6667 | 0.5778 | 0.4360 | 0.5185 | 0.4737 | 0.4082 | 0.6897 | 0.5128 |
| Support Vector Classifier | 0.5542 | 0.5897 | 0.5714 | 0.7215 | 0.7362 | 0.7288 | 0.8333 | 0.6897 | 0.7547 |
| Gaussian Naive Bayes | 0.5258 | 0.9808 | 0.6846 | 0.7891 | 0.3487 | 0.4837 | 0.8710 | 0.4655 | 0.6067 |
| Logistic Regression | 0.5550 | 0.6795 | 0.6110 | 0.7142 | 0.7491 | 0.7312 | 0.9024 | 0.6379 | 0.7475 |

| **Model** | Precision dataset (a) | Recall dataset (a) | F1-score dataset (a) | Precision dataset (b) | Recall dataset (b) | F1-score dataset (b) | Precision dataset (c) | Recall dataset (c) | F1-score dataset (c) |
|---|---|---|---|---|---|---|---|---|---|
| Meta Classifier | 0.5990 | 0.7564 | 0.6686 | 0.7335 | 0.7288 | 0.7311 | 0.9545 | 0.7241 | 0.8235 |

Table 4.3: Precision, Recall and F1-Score of Each Predictive Model

# Chapter 5

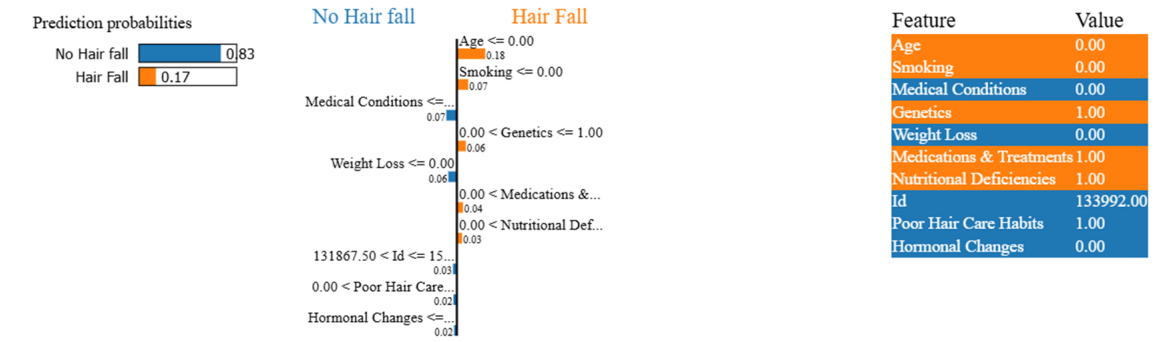
# Integrated Model Evaluation and Performance Analysis

This section explains feature selection, data imputation, model evaluation and improvement. LIME and SHAP were applied to extract important features, Missing variables were treated with predicted imputation; validation of our accuracy happened through the Mean Squared Error (MSE). After imputing values, the three datasets were merged into one based on retaining only the most influential features. Furthermore, we implemented predictive models again Thus, for assessing the influence of attributes upon the performance of these models prediction, accuracy, f1-score, and recall were put together in evaluating the performance of these models. More accuracy was obtained using a tuned-model 1D Convolutional Neural Network(CNN) with an autoencoder. The improved classification capability for the fine-tuned model was then put to the test in comparison with the conventional models of prediction. Finally, to illustrate the importance of relevant attributes and the performance of different model architectures, we carefully reflected on the results.

## 5.1 LIME Explanation

Examining feature relevance for Hair Fall prediction, the following figure 5.1 demonstrates LIME (Local Interpretable Model-Agnostic Explanations) visualizations for three distinct datasets, labeled as (a), (b), and (c). LIME highlights the most important features for each instance so that we can comprehend the contributions of individual features to the predictions of the model. With important influencing features including 'Age', 'Smoking', 'Genetics', 'Medications & Treatments', and 'Nutritional Deficiencies', the model predicts No Hair Fall (83%) and Hair Fall (17%), in dataset (a). These features positively (orange-colored) influence hair fall predictions based on figure 5.1. The prediction probabilities in dataset (b) are more balanced (No Hair Fall 60%, Hair Fall 40%), and the primary impacting features are 'Job_role', 'Smoking', 'Stress', 'Salary' and 'Province', thereby demonstrating that both biological and socioeconomic elements influence hair fall. On the other hand, dataset (c) highly predicts No Hair Fall (97%) with minor impact from features like 'Pressure_level', 'School_assessment', 'Hair_washing' and 'Swimming', therefore highlighting how lifestyle habits alongside external factors can assist to determine hair health. These above LIME insights indicate the behavior of the model understandable, therefore clarifying the locally significant features in predicting hair fall

across each dataset.




(a)


(b)


(c)

Figure 5.1: Dataset (a) LIME Feature Values, Dataset (b) LIME Feature Values and Dataset (c) LIME Feature Values

## 5.2 SHAP Explanation

In our research, we deployed SHAP (SHapley Additive exPlanations) to look into machine learning prediction output interpretation and corresponding feature importances across these three datasets: (a), (b) and (c). SHAP values show the importance of each feature in the prediction of the model while taking into view all possible combinations of features and evaluating how adding each feature changed output in terms of the alternative predictions. In figure 5.2, we find strongly contributing factors to hair fall by concentrating on the positive and negative x-axis of the SHAP value distribution. With features exceeding a threshold of 0.5 considered as having a moderate to high influence, SHAP values show how much each feature pushes the prediction of the model towards hair fall. We especially examine features with strong SHAP values shown by red dots on both sides of the x-axis. By filtering out weakly contributing elements, this threshold-based method lowers noise and increases both model efficiency and interpretability. In Dataset (a), the most significant features crossing the 0.5 SHAP threshold appearing as red dots with high values include 'Genetics', 'Smoking', 'Weight Loss', "Medications & Treatments', 'Medical Conditions', 'Poor Hair Care Habits', 'Hormonal Changes', 'Stress' and 'Age' across both the sides of x-axis according to the figure 5.2. In addition, dataset (b), features exceeding the 0.5 SHAP threshold marked as dots in red with high values, captured 'Job_role', 'Age', 'Genetics', 'Smoking', 'Stress', 'Salary', 'Gender', 'Province' as well as 'Education' on positive and negative sides of the x-axis. Finally, dataset (c) has red dotted high values for 'Libido', 'Stress', 'Staying_up_late', 'Hair_washing', 'Hair_grease', and 'Job_role' on both sides of the x-axis, major influential features exceeding the 0.5 SHAP value. Moreover, with this SHAP-based feature selection approach, only those hair loss causing factors deemed very relevant are considered while taking into account the risk factors, to the overall improvement of the model.

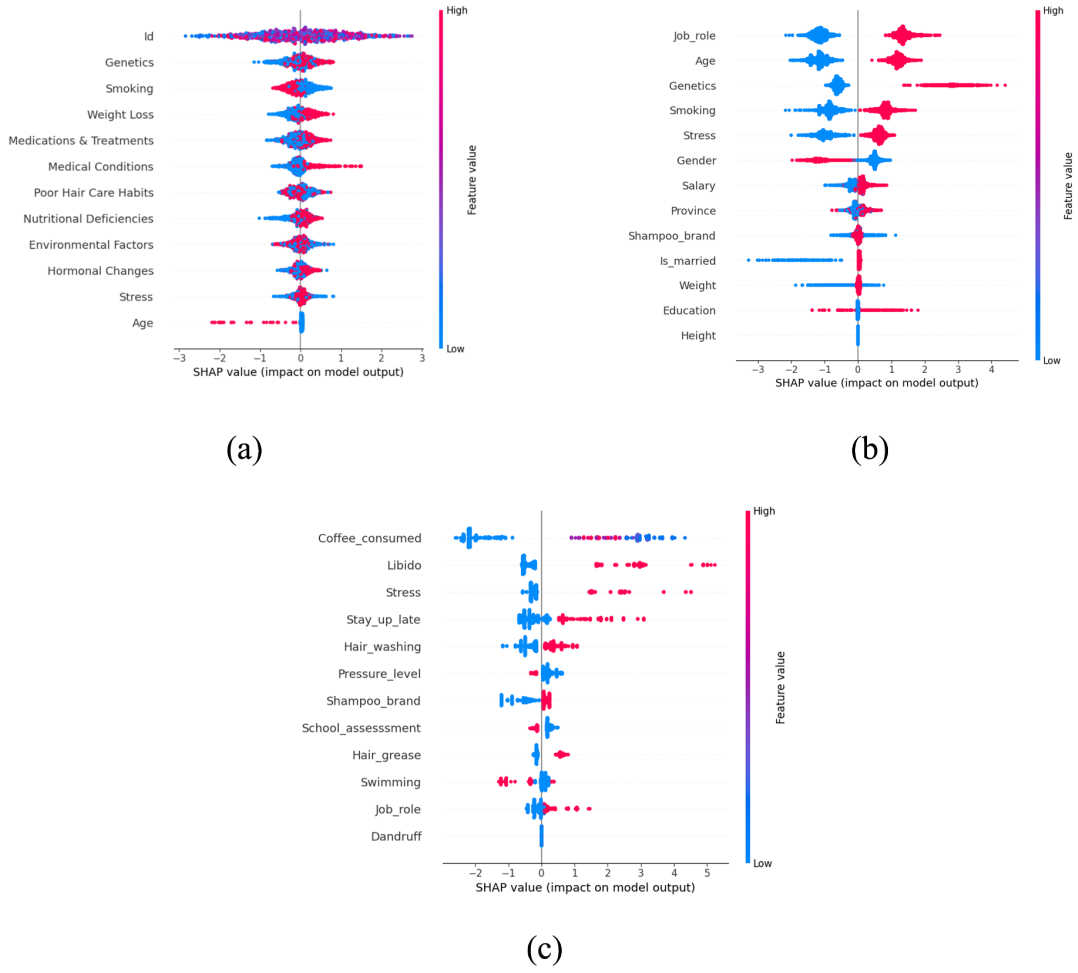



Figure 5.2: Dataset (a) SHAP Feature Values, Dataset (b) SHAP Feature Values and Dataset (c) SHAP Feature Values

## 5.3 Feature Selection and Extraction

Feature selection is a very crucial step in our research to ensure that the predictive model includes only attributes whose importance has been established while minimizing noise and redundancy. Our strategy integrates numerous interpretability techniques, including, LIME (Local Interpretable Model-agnostic Explanations) in figure 5.1, SHAP (Shapley Additive Explanations) in figure 5.2 and correlation analysis in figure 4.1, to make strengthened decisions on feature inclusion and expulsion. SHAP (Shapley Additive Explanations) was applied to analyse global feature importance across all datasets, which provides a full understanding of how each parameter affects model predictions. By contrast, LIME (Local Interpretable Model-agnostic Explanations) provided instance-level insights through the use of local feature importance. SHAP was given higher priority than LIME in the last choice phase since our objective is to generalise the results across datasets instead of emphasising on individual predictions. We performed correlation analysis, retaining solely those features showing a statistically significant correlation with the target feature Hair Loss, hence validating SHAP's observations. This combined technique ensured that

selected features possessed both statistical significance and predictive value. During the selection process, 'Age' was eliminated due to its inconsistent impact across different datasets, as SHAP values varied between positive and negative correlations. This variability indicated that 'Age' might be dataset-dependent rather than a stable predictor of hair loss. While a stable feature assists the model to learn meaningful patterns, an unstable feature might generate noise. On the other hand, 'Smoking' showed inverse SHAP values over several datasets yet maintained a constant relationship with the target feature Hair Loss. Its inclusion as a consistent predictive feature was justified by this consistency in correlation. Features that showed weak SHAP impact but strong correlation were carefully evaluated to prevent unnecessary noise in the model. Based on the above structured feature selection process, the final impactful selected features were identified across three datasets.

- Dataset (a): 'Weight Loss', 'Medications & Treatments', 'Genetics', 'Stress', 'Nutritional Deficiencies'.
- Dataset (b): 'Stress', 'Smoking', 'Salary', 'Province'.
- Dataset (c): 'Hair_grease', 'Hair_washing', 'Libido', 'Stress', 'Staying_up_late', 'Job_role'.

High SHAP significance, correlation with Hair Loss, and consistency across several datasets assisted determine the selection of these specific features. This approach enhances model interpretability, improves generalisability, and assures that only the most impactful features contribute to the final predictive model.

## 5.4 Feature Imputation

To ensure a common feature set across dataset (a), dataset (b), and dataset (c), a structured imputation strategy based on the identified impactful features from section 5.3 was implemented so that only those impactful features could be transferred among the datasets. The predictive imputation process was implemented to be systematic such that there were statistical consistencies and great reductions in biases. The first step was to split the dataset holding the missing feature at a ratio of 70:30. The next step was to also split the source dataset with similar ratio indices so that it would maintain the alignment with the target dataset. Hereafter, a Random Forest Classifier model was fed with the resultant 70% augmented balanced portion of the source dataset for training that model to capture complex feature interactions while ensuring robust prediction accuracy. This trained model was then used to impute the missing feature in that 70% portion of the target dataset. A K-Nearest Neighbors (KNN) model was then trained for imputation of the remaining part on the specific 30% actual test data portion of the source dataset. The use of the KNN for the 30% imputation ensured local variability, preventing the test data from inheriting identical patterns from the training data. Therefore, This two-step imputation process ensured that the imputed values remained statistically meaningful, reducing biases while preserving natural data distributions. To evaluate the reliability of the imputed features, Mean Squared Error (MSE) was calculated for each imputed feature, with values ranging between 0.3 and 0.5 shown in figure 5.3, indicating accurate and reliable imputation. This imputation strategy helped to develop a solid and comprehensive datasets, from which Hair Loss prediction would be

enhanced without bias and data leakage. At first, the imputing process started with finding the source dataset of each missing feature so that data for imputation would be taken only from the most strongly correlated and SHAP values yielding dataset. Thus, 'Genetics' was imputed from dataset (a) to dataset (c) because dataset (b) has 'Genetics' but showed no strong association to 'Hair Loss'. 'Weight Loss', 'Medications & Treatments', and 'Nutritional Deficiencies' were imputed from dataset (a) to dataset (b) and dataset (c) because they were missing from both datasets and had strong relevance to 'Hair Loss' in dataset (a). 'Smoking' was imputed from dataset (b) into dataset (c) since that was already present in both datasets (a) and (b), but was more correlated and had higher SHAP values for dataset (b) which made it the more reliable source. Similarly, 'Job_role' was imputed in dataset (a) using dataset (c), as this dataset had this feature yet showed a greater relationship with 'Hair Loss'. 'Hair_washing' and 'Hair_grease' were imputed to dataset (a) and dataset (b) from dataset (c) because they were only present in dataset (c) that contributed a strong relation to 'Hair Loss'. 'Salary' and 'Province' were considered to be imputed to dataset (a) and dataset (c) using dataset (b), while 'Libido' and 'Staying_up_ late' were considered to be imputed to dataset (a) and dataset (b) using dataset (c) so that all datasets would have a comprehensive and meaningful feature set. On the other hand, 'Stress' was identified as the common feature among all three databases, thus not necessitating imputation. This imputation strategy helped to develop a solid and comprehensive datasets, from which hair loss prediction would be enhanced without bias and data leakage.

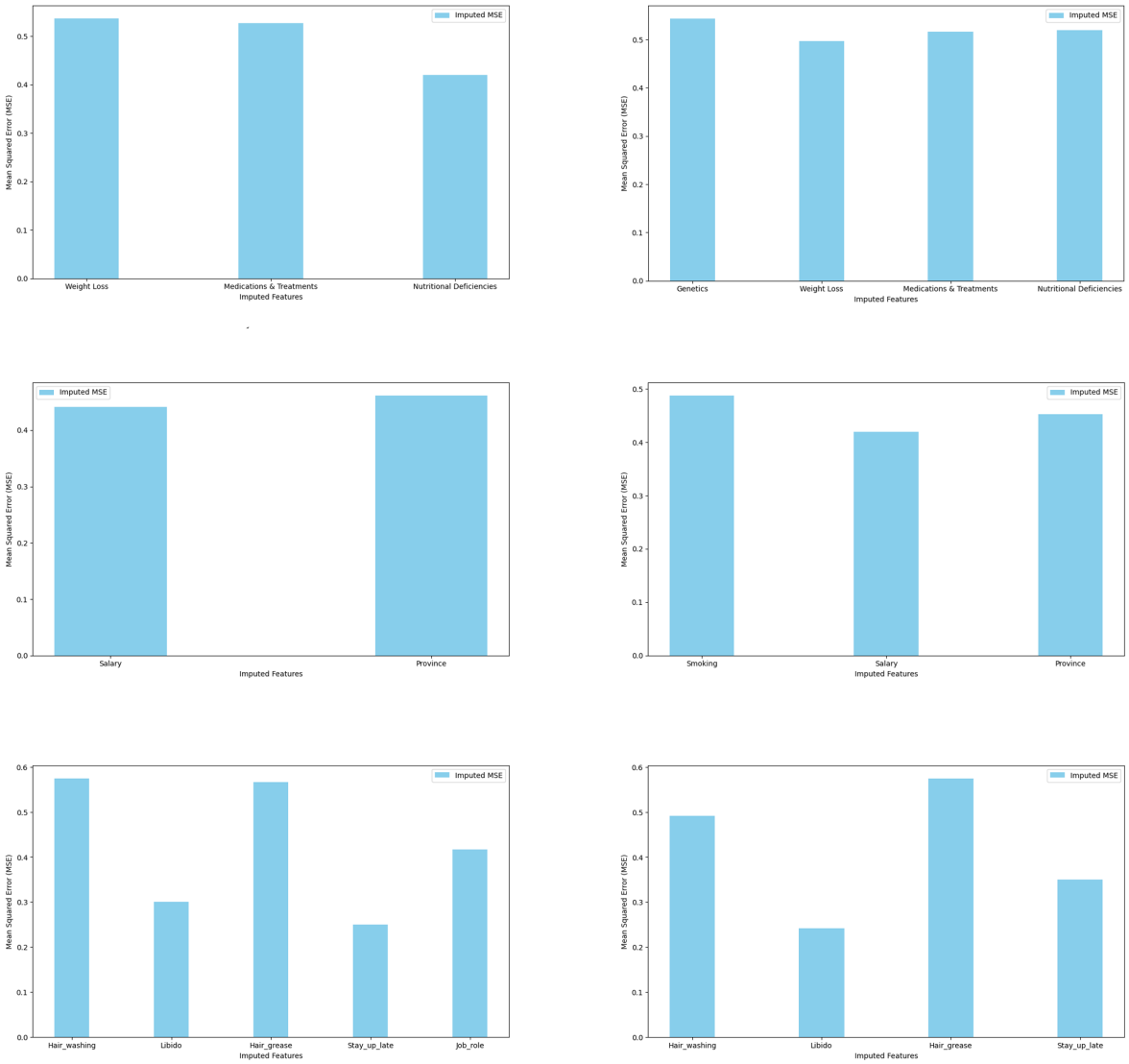



Figure 5.3: Mean Squared Error of Imputed Features

### 5.4.1 Dataset Merging

After performing feature selection (5.3) and feature imputation (5.4), all three datasets were aligned to a common feature set, including the target feature 'Hair Loss'. The finalized feature set comprised 'Genetics', 'Stress', 'Province', 'Salary', 'Hair_washing', 'Nutritional Deficiencies', 'Smoking', 'Job_role', 'Libido', 'Medications & Treatments', 'Stay_up_ late', 'Weight Loss', 'Hair_grease', and the target feature 'Hair Loss'. At this point, a merging approach has been performed using 'Column-Consolidation' with 'Row-Wise Concatenation' including the common feature set across three distinct datasets for constructing a comprehensive and consistent merged dataset. This approach ensured that all relevant features were preserved while incorporating diverse instances from each dataset. The resulting merged dataset contained 14 features including the target feature 'Hair Loss' and a total of

9,316 rows. Consolidating data from three datasets assisted the merged dataset to provide a more broader and balanced representation of the data distribution, hence lowering data leakage and associated biases.

## 5.5 Model Comparison and Evaluation

This section looks into how the predictive models perform before and after dataset fusion using impact measurements of accuracy, F1-score, recall, and precision. A comparative study was initiated to compare the model performances before and after dataset merging and to use the results in tuning a 1D Convolutional Neural Network (CNN) with an autoencoder mixture toward maximizing the accuracy. A further evaluation was done through the comparison of test loss, accuracy, and F1-score of the model. Finally, we have discussed the reasoning for selecting tuned model and how they serve well using the extracted feature dataset to enhance performance.

### 5.5.1 Impact of Merged Dataset on Predictive Model Performance

After the initial application of predictive models such as XGBoost Classifier, Suport Vector Classifier, Decision Tree Classifier, Logistic Regression, Gaussian Naive Bayes, Random Forest Classifier, Meta Classifier, and K-nearest Neighbors on the three different datasets, we proceeded to merge them in an attempt to enhance their predictive performances. The merging procedure involved integrating the datasets into one single unified merged dataset through fourteen common features, including the target feature. Thereafter, the same predictive models that were applied on the merged dataset in section 5.4.1 were used to obtain accuracy, F1-score, precision, and recall so we will analyze the impact of the merged dataset on model performance by comparing these performance metrics against those derived from the initial independent three datasets. The discussion will detail how the merged dataset affects the model's capacity to generalize and provide accurate predictions while outlining all possible advantages and disadvantages of merging datasets during predictive modeling..

| **Model Name** | **Accuracy** Percentage | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| K-Nearest Neighbors | 88.62% | 0.8909 | 0.8528 | 0.8715 |
| Random Forest Classifier | 88.30% | 0.8800 | 0.8584 | 0.8690 |
| Decision Tree Classifier | 88.26% | 0.8817 | 0.8552 | 0.8683 |
| XGBoost Classifier | 88.26% | 0.8817 | 0.8552 | 0.8683 |
| Support Vector Classifier | 88.09% | 0.8875 | 0.8560 | 0.8666 |
| Gaussian Naive Bayes | 74.45% | 0.7580 | 0.6392 | 0.6936 |

| Model Name | Accuracy Percentage | Precision | Recall | F_1 Score |
|---|---|---|---|---|
| Logistic Regression | 85.72% | 0.8531 | 0.8267 | 0.8397 |
| Meta Classifier | 88.69% | 0.8937 | 0.8513 | 0.8720 |

Table 5.1: Summary of Predictive Model Performance on Merged Dataset

**K-Nearest Neighbors:** The results indicated in table 5.1 show that the K-nearest neighbors now achieved an accuracy of 88.62%, whereas in table 4.2, before merging of datasets, the accuracies of KNN were 48.33%, 48.11%, and 27.50% for datasets (a), (b), and (c), respectively. Table 5.1 further presents the improvement of precision, recall, and f1-score at 0.8909, 0.8528, and 0.8715, respectively, in comparison to the values in Table 4.3 for the three initial datasets.

**Random Forest Classifier:** The accuracy of the Random Forest Classifier has reached an impressive 88.30% as depicted in table 5.1 for the merged dataset, as against the significant reductions in accuracy, showing 45.33%, 45.75%, and 30.83%, respectively, for each of the three initial datasets in table 4.2. Thus, as demonstrated further in table 5.1, the precision, recall, and f1-score have also risen to 0.8800, 0.8584, and 0.8690 compared to the precision, recall, and f1-score from table 4.3 for the individual datasets.

**Decision Tree Classifier:** According to table 4.2, the Decision Tree Classifier accuracy occasionally attained values of 48.00%, 47.35%, and 32.50% among the three initial datasets. Nonetheless, according to table 5.1, the improvement of Decision Tree Classifier had been significant and reached an accuracy of 88.26%. This improvement is translated into precision, recall, and f1 Score in table 5.1, which shows 0.8817, 0.8552, and 0.8683 in contrast to the numbers in table 4.3 for the three different datasets.

**XGBoost Classifier:** Table 5.1 shows that the accuracy of the XGBoost Classifier has further increased to 88.26%, while in table 4.2, it was 49.33%, 47.43%, and 36.67% in the case of the original datasets. From table 5.1, the precision, recall, and f1-score, respectively, of the XGBoost Classifier for this table increased to 0.8817, 0.8552, and 0.8683 in contrast to table 4.3.

**Support Vector Classifier:** From table 5.1, we see that the Support Vector Classifier has increased its accuracy to 88.09% compared to 54.00%, 75.00%, and 78.33% in table 4.2. Furthermore, precision, recall, and f1-score values in table 5.1 are improved for table 4.3 to 0.8875, 0.8560, and 0.8666, respectively.

**Gaussian Naive Bayes:** According to table 5.1 and table 4.2, the accuracy of Gaussian Naive Bayes has improved to 74.45% for the merged dataset, compared to 53.00% for dataset A and 66.04% for dataset (b). Slightly lower, though, from that of dataset (c) in table 5.1 and table 4.2, for which the accuracy was 70.83%. The precision, recall, and f1-score in table 5.1 have slightly dropped to 0.7580, 0.6392, and 0.6936, respectively, compared to those represented in table 4.3.

**Logistic Regression:** From tables 5.1and 4.2, we observe that the accuracy of Logistic Regression has increased from 55.00% (dataset a) to 85.72% for the merged dataset. However, in comparison to datasets (b) and (c) in table4.2, where the accuracy was 74.87% and 79.17%, the accuracy of Logistic Regression has increased. Moreover, the precision, recall, and f1-score in table 5.1 are given as 0.8531, 0.8267, and 0.8397, respectively, with a slight change from the values in table 4.3.

**Meta Classifier:** The Meta Classifier's accuracy increased to 88.69% for the merged dataset as presented in table 5.1, compared to a table 4.2 accuracy of 61.00%, 75.55%, and 85.00% for the three initial datasets. Also, it is visible from table 5.1 and table 4.3 that precision, recall, and f1-score improved to 0.8937, 0.8513, and 0.8720, respectively, when compared to the individual datasets.

### 5.5.2 Model Tuning and Selection

When we used the merged dataset, there was increase in accuracy,precision,recall and f1-score which can be observed in section 5.5.1, however we still wanted to increase our accuracy so we introduced a tuning model which was 1D CNN with autoencoder to optimize the performance metric values of the accuracy ,precision recall and f1-score respectively

| **Model Name** | **Parameters** | **Data Enhancement** |
|---|---|---|
| 1D CNN with Autoencoder | epochs = 100,<br>batch_size = 64,<br>validation_split = 0.3,<br>encoding dimension = 8,<br>filters (Conv1D) =32, 64,<br>kernel_size = 2,<br>dropout = 0.3, 0.5,<br>early_stopping = 10,<br>learning_rate (adam) = 0.0001,<br>input_shape=<br>(x_train_encoded.shape[1], 1),<br>autoencoder (epochs = 100,<br>batch_size = 256 | StandardScaler |

Table 5.2: Summary of HyperTuning Parameters

**1D CNN (Convolutional Neural Network) with Autoencoder:** The first step in the entire machine learning prediction process is Data preprocessing, which includes standardizing the dataset and separating training and testing sets. An independent Autoencoder was used which compresses the input into an 8-dimensional latent representation for dimensionality reduction and feature extraction. Autoencoder training is performed using the MSE loss function, enabling one to recover essential characteristic patterns. The compressed feature representations extracted from the autoencoder are fed as input into Convolutional Neural Networks (CNN) for final prediction. The CNN has convolutional layers with batch normalization and dropout layers used in stabilizing training and preventing the system from overfitting. It has a final dense layer to give the final CNN output, and then a sigmoid activation function to put it through binary classification. Adam optimizer was also used to optimize the model in the direction of binary cross-entropy loss, and early stopping, reducing learning rate, and checkpointing were some of the techniques undertaken to avoid overfitting, obtaining the best possible performance . Emphasis is also laid on the fact that the benefits of dimensionality reduction, to preserve

the important information, in analyzing the highly complex structure concerning predicting hair loss have been merged through the autoencoder with CNN

### 5.5.3 Tuned-Model Performance Visualization and Comparison

The following subsections will describe the model tuning process for the 1D Convolutional Neural Network with autoencoder (CNN). Various hyperparameter combinations were experimented for the model. The resultant effect of these tuning parameters on model performance is discussed in terms of test accuracy and model loss, where relevant visualizations showing the metrics over different parameter combinations.

**1D CNN (Convolutional Neural Network) with Autoencoder Accuracy:** In figure 5.4, the accuracy measured for 100 epochs on training and validation data is reported for the 1D CNN with Autoencoder. The training accuracy evolves from 0.60 at the beginning, increasing sharply in the first 20 epochs, and gradually stabilizing around 0.92 in the last few epochs. The validation accuracy aligns with this trend, starting at around 0.73, steadily increasing over the epochs, and finally reaching around 0.95 later in the training set. As we can observe that the validation accuracy remain above the training accuracy we can come to a conclusion that the model is generalization the data well enough without overfitting it.

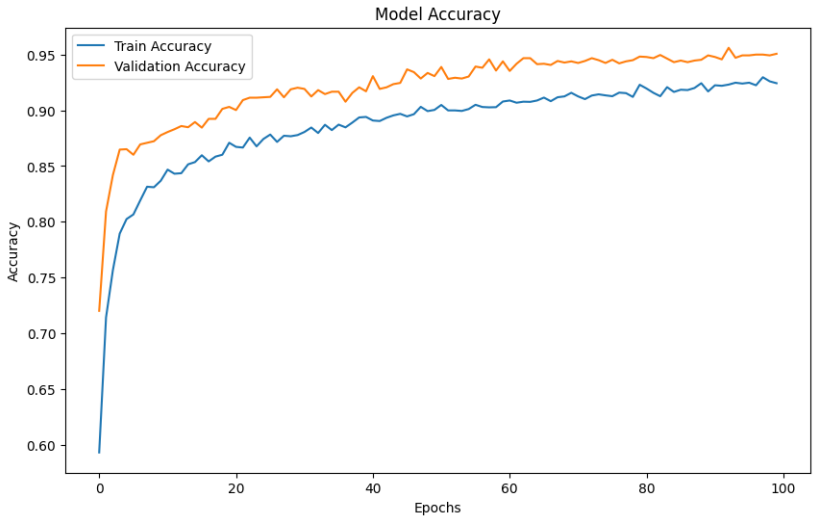



Figure 5.4: 1D CNN (Convolutional Neural Network) with Autoencoder Accuracy

**1D CNN (Convolutional Neural Network) with Autoencoder Loss:** As shown in figure 5.5, the loss curves for both training and validation data are plotted against 100 epochs. The training loss started at about 1.8, steeply decreased for the first 20 epochs, and then stabilized at approximately 0.2 towards the end. Moreover, we can also observe that validation loss had a similar decreasing trend, remaining slightly below the training loss through all epochs. we can conclude that there is a huge consistent reduction in the loss value which shows that the model had a smooth minimization of error and had better stability of learning.

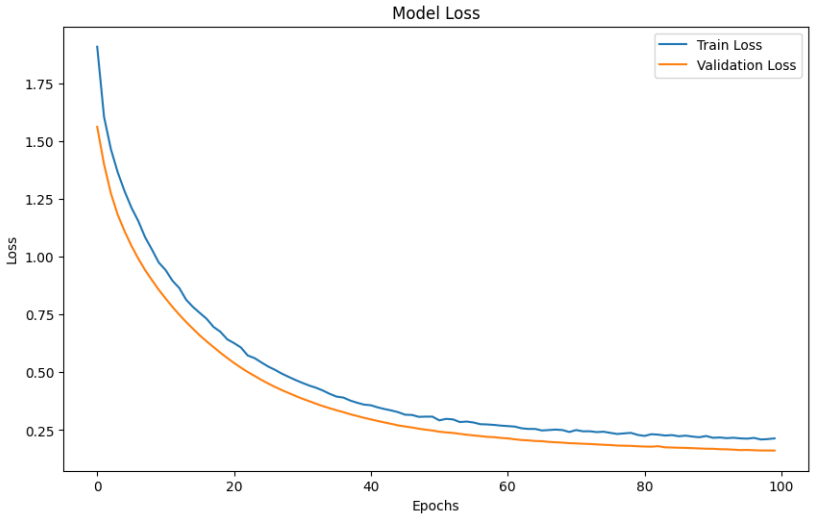



Figure 5.5: 1D CNN (Convolutional Neural Network) with Autoencoder Loss

| **Model Name** | **Test Accuracy** Percentage | **Test Loss** Percentage | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|---|
| 1D CNN with Autoencoder | 96.78% | 13.65% | 0.9574 | 0.9737 | 0.9655 |

Table 5.3: Summary of 1D CNN with Autoencoder Model Performance Metrics

The 1D CNN with Autoencoder presented in table 5.3 shows a high efficiency with a test accuracy of 96.78% and a low test loss of 0.1364, thus demonstrating its excellent generalization. It has fairly satisfactory precision at 0.9574, indicating very few false positives; the recall rate of 0.9737 means that almost all positives were detected correctly. The F1-score of 0.9655 indicates a great balance between precision and recall, thus giving assurance of the credibility of the model. Therefore, we can conclude that the 1D CNN with Autoencoder is considered an efficient classification model, ensuring accuracy and consistency in its prediction.

### 5.5.4 Results and Discussions

Predictive models performance was much improved by the combined dataset when compared to the initial three distinct datasets. Limited feature space and missing values across three initial datasets caused predictive traditional models to demonstrate rather poor performance metrics before dataset merging. Each dataset contained distinct but incomplete sets of features, leading to inconsistencies in model training. By identifying the impactful features (5.3) and imputing the missing features (5.4) across dataset (a), dataset (b) and dataset (c), the datasets were transformed into a unified structure with a common feature set. This imputation process enhanced data completeness and consistency, ensuring that all datasets contributed valuable information to the merged dataset (5.4.1). As a result, the predictive models trained on the merged dataset demonstrated improved accuracy, precision, recall, and f1-score compared to their performance on individual datasets according to the

model performance table of 5.1. The integration of the merged dataset (5.4.1) reduced the information gap and provided a more comprehensive representation of the factors influencing hair loss, enabling the models to generalize better.Furthermore, a 1D Convolutional Neural Network (CNN)-based autoencoder was fine-tuned and trained on the merged dataset, recording the best-performing metrics across all models demonstrated in table 5.3. The autoencoder effectively captured latent representations within the data, helping to preserve essential patterns while minimizing noise. Model performance gains signify the success of feature imputation, dataset integration, and machine learning methods in resolving issues of data sparsity and better predicting processes. The merged dataset was built up with a well-distributed set of impactful features identified through model interpretability techniques, so it could constitute a very powerful basis for training machine learning models which ultimately resulted in performance improvements in terms of reliability and accuracy at output.

# Chapter 6

# Conclusion

Alopecia Areata is a very prevalent issue globally, often leading to emotional suffering and reduced self-esteem. This paper presents a machine learning-based approach for the prediction of hair loss with the involvement of numerical data. According to our research, 'Hair loss' analysis greatly benefited from feature imputation and dataset merging as ways of improving the power of prediction. Models trained on separate datasets performed poorly at first due to incomplete features and limited data representation. The influential features identified by SHAP, and correlation analysis across three initial datasets allowed us to impute the missing values with the common feature set and make the merged dataset more comprehensive for training with models. Moreover, traditional predictive machine-learning models including K-Nearest Neighbours, Support Vector Classifier, Random Forest Classifier, Decision Tree Classifier, Logistic Regression, Gaussian Naive Bayes, XGBoost Classifier and Meta Classifier, performed well, while tuned 1D CNN with an autoencoder scored best, capturing the highest interaction of complex features on the merged dataset.These findings highlight the need of structured feature engineering and dataset integration in enhancing hair loss prediction.

## 6.1 Limitations and Future Work

**Limitations:** Because of privacy concerns, a number of dermatologists and clinics were unwilling to share patient data, thus preventing us from accessing local data. Afterwards, we resorted to foreign datasets.
**Future Work:** In future work, we plan to incorporate an image dataset consisting of scalp images taken over multiple years. This will allow us to analyze the images for features that influence hair loss detection. Additionally, we aim to enhance our dataset by integrating high-quality images and numerical data from hospitals, enabling more thorough feature extraction. By combining these new features with those already used in our models, and further optimizing the models, we expect to achieve more accurate predictions. This approach will also help us better understand how these features impact model performance, ultimately leading to more reliable tools for hair loss diagnosis and prediction.

# Bibliography


[1] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–242, 1958. [Online]. Available: https://www.jstor.org/stable/2983890.

[2] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. DOI: 10.1109/TIT.1967.1053964.

[3] C. Cortes and V. Vapnik, "Support-vector networks," en, *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. DOI: 10.1007/BF00994018.

[4] L. Breiman, *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1007/BF00116251.

[5] M. Buscema, S. Terzi, and W. Tastle, "A new meta-classifier," Jul. 2010, pp. 1–7, ISBN: 978-1-4244-7859-0. DOI: 10.1109/NAFIPS.2010.5548298.

[6] I. Park and S. Lee, "Spatial prediction of landslide susceptibility using a decision tree approach: A case study of the pyeongchang area, korea," *International Journal of Remote Sensing*, vol. 35, no. 16, pp. 6089–6112, 2014.

[7] A. Vujovic and V. Del Marmol, "The female pattern hair loss: Review of etiopathogenesis and diagnosis," en, *BioMed Research International*, vol. 2014, pp. 1–8, 2014, ISSN: 2314-6133. DOI: 10.1155/2014/767628. [Online]. Available: http://dx.doi.org/10.1155/2014/767628.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785.

[9] I. Kapoor and A. Mishra, "Automated classification method for early diagnosis of alopecia using machine learning," *Procedia Computer Science*, vol. 132, pp. 437–443, 2018, International Conference on Computational Intelligence and Data Science, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.157. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308913.

[10] H. Benhabiles, K. Hammoudi, Z. Yang, *et al.*, "Deep learning based detection of hair loss levels from facial images," in *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2019, pp. 1–6. DOI: 10.1109/IPTA.2019.8936122.

[11] X. Li *et al.*, "Evaluation of 1d cnn autoencoders for lithium-ion battery state of health estimation," in *PHM Society Conference*, 2019. DOI: 10.36001/phmconf.2019.876. [Online]. Available: https://doi.org/10.36001/phmconf.2019.876.

[12] W.-J. Chang, L.-B. Chen, M.-C. Chen, Y.-C. Chiu, and J.-Y. Lin, "Scalpeye: A deep learning-based scalp hair inspection and diagnosis system for scalp health," *IEEE Access*, vol. 8, pp. 134 826–134 837, 2020.

[13] S. Lee, J. W. Lee, S. J. Choe, *et al.*, "Clinically applicable deep learning framework for measurement of the extent of hair loss in patients with alopecia areata," en, *JAMA Dermatology*, vol. 156, no. 9, p. 1018, Sep. 1, 2020, ISSN: 2168-6068. DOI: 10.1001/jamadermatol.2020.2188. [Online]. Available: http://dx.doi.org/10.1001/jamadermatol.2020.2188.

[14] F. Z. Okwonu, B. L. Asaju, and F. I. Arunaye, "Breakdown analysis of pearson correlation coefficient and robust correlation methods," *IOP Conference Series: Materials Science and Engineering*, vol. 917, p. 012 065, 2020. DOI: 10.1088/1757-899X/917/1/012065. [Online]. Available: https://doi.org/10.1088/1757-899X/917/1/012065.

[15] V. Subramanian and R. Kumar, "Binary cross entropy with deep learning technique for image classification," *ResearchGate*, 2020. [Online]. Available: https://www.researchgate.net/publication/344854379.

[16] J. Rala Cordeiro, A. Raimundo, O. Postolache, and P. Sebastião, "Neural architecture search for 1d cnns—different approaches tests and measurements," *Sensors*, vol. 21, no. 23, 2021, ISSN: 1424-8220. DOI: 10.3390/s21237990. [Online]. Available: https://www.mdpi.com/1424-8220/21/23/7990.

[17] S. Sa, "Comparative study of naive bayes, gaussian naive bayes classifier and decision tree algorithms for prediction of heart diseases," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, pp. 475–486, Mar. 2021. DOI: 10.22214/ijraset.2021.33228.

[18] S. Aditya, S. Sidhu, and M. Kanchana, "Prediction of alopecia areata using machine learning techniques," in *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, 2022, pp. 1–6. DOI: 10.1109/ICDSIS55133.2022.9915804.

[19] "An analysis of alopecia areata classification framework for human hair loss based on vgg-svm approach," *Journal of Pharmaceutical Negative Results*, vol. 13, no. S01, Jan. 1, 2022. DOI: 10.47750/pnr.2022.13.s01.02. [Online]. Available: http://dx.doi.org/10.47750/pnr.2022.13.S01.02.

[20] F. Khatun, M. R. Ajmain, S. A. Khushbu, N. J. Ria, and S. R. H. Noori, "Survey-based machine learning approaches to diagnosis of hair fall disorder in bangladeshi community," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, Oct. 3, 2022. DOI: 10.1109/icccnt54827.2022.9984226. [Online]. Available: http://dx.doi.org/10.1109/icccnt54827.2022.9984226.

[21] J.-H. Kim, S. Kwon, J. Fu, and J.-H. Park, "Hair follicle classification and hair loss severity estimation using mask r-cnn," *Journal of Imaging*, vol. 8, no. 10, 2022, ISSN: 2313-433X. DOI: 10.3390/jimaging8100283. [Online]. Available: https://www.mdpi.com/2313-433X/8/10/283.

[22] M. Kim, S. Kang, and B.-D. Lee, "Evaluation of automated measurement of hair density using deep neural networks," en, *Sensors*, vol. 22, no. 2, p. 650, Jan. 14, 2022, ISSN: 1424-8220. DOI: 10.3390/s22020650. [Online]. Available: http://dx.doi.org/10.3390/s22020650.

[23] S. Sayyad and D. Midhunchakkaravarthy, "Deep study on alopecia areata diagnosis for hair loss related autoimmune disorder problem using machine learning techniques," en, *SSRN Electronic Journal*, 2022, ISSN: 1556-5068. DOI: 10.2139/ssrn.4054490. [Online]. Available: http://dx.doi.org/10.2139/ssrn.4054490.

[24] W. W. Heng and N. A. Abdul-Kadir, "Deep learning and explainable machine learning on hair disease detection," in *2023 IEEE 5th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, 2023, pp. 150–153. DOI: 10.1109/ECBIOS57802.2023.10218472.

[25] M. Kim, Y. Gil, Y. Kim, and J. Kim, "Deep-learning-based scalp image analysis using limited data," en, *Electronics*, vol. 12, no. 6, p. 1380, Mar. 14, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12061380. [Online]. Available: http://dx.doi.org/10.3390/electronics12061380.

[26] J. Madke, M. Sondur, and S. Bhatlawande, "Alopecia pattern detection in males using classical machine learning," 2023 International Conference on Inventive Computation Technologies (ICICT), IEEE, Apr. 26, 2023. DOI: 10.1109/icict57646.2023.10134212. [Online]. Available: http://dx.doi.org/10.1109/ICICT57646.2023.10134212.

[27] M. Roy and A. T. Protity, "Hair and scalp disease detection using machine learning and image processing," *arXiv*, 2023. DOI: 10.48550/ARXIV.2301.00122. [Online]. Available: https://arxiv.org/abs/2301.00122.

[28] C. Saraswathi and B. Pushpa, "Frcnn based deep learning for identification and classification of alopecia areata," in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2023, pp. 1–7. DOI: 10.1109/ICECCT56650.2023.10179804.

[29] C. Saraswathi and B. Pushpa, "Multi-class support vector machine classification for detecting alopecia areata and scalp diseases," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 11s, pp. 63–70, Oct. 7, 2023, ISSN: 2321-8169. DOI: 10.17762/ijritcc.v11i11s.8071. [Online]. Available: http://dx.doi.org/10.17762/ijritcc.v11i11s.8071.

[30] J. Xiong, G. Chen, Z. Liu, *et al.*, "Construction of regulatory network for alopecia areata progression and identification of immune monitoring genes based on multiple machine-learning algorithms," en, *Precision Clinical Medicine*, vol. 6, no. 2, May 22, 2023, ISSN: 2096-5303. DOI: 10.1093/pcmedi/pbad009. [Online]. Available: http://dx.doi.org/10.1093/pcmedi/pbad009.

[31] R. Behal, P. Priya, Yuvraj, A. Kapoor, and A. Mishra, "Hair loss stage prediction using deep learning," in *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, vol. 5, 2024, pp. 1263–1267. DOI: 10.1109/IC2PCT60090.2024.10486442.

[32] W. Xu, D. Wei, and X. Song, "Identification of slc40a1, lcn2, creb5, and slc7a11 as ferroptosis-related biomarkers in alopecia areata through machine learning," en, *Scientific Reports*, vol. 14, no. 1, Feb. 15, 2024, ISSN: 2045-2322. DOI: 10.1038/s41598-024-54278-4. [Online]. Available: http://dx.doi.org/10.1038/s41598-024-54278-4.