Koita Centre for Digital Health
Indian Institute of Technology, Bombay

A report for the first phase submitted in partial fulfilment of the requirements for completion of the Dual Degree project

Non-Invasive Detection of Anemia Using Anatomic Images

Submitted by

Shrey Gupta (190100112)

Guide: Prof. Nirmal Punjabi (KCDH)

Co-Guide: Prof. Ganesh Ramakrishnan (CSE)

(October 2023)

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shrey Gupta

190100112

Date: 13th October 2023

List of Figures

List of Tables

Table of Contents

1. Abstract

Anemia, a condition characterized by low hemoglobin levels, affects approximately 1.6 billion people worldwide. This study investigates non-invasive methods for anemia detection, comparing their performance in both classification and regression tasks. Traditional approaches rely on invasive blood tests, but recent developments have focused on non-invasive techniques using anatomical images like the eye, palm, nail, tongue, and conjunctiva to estimate hemoglobin levels.

The research methodology involves acquiring a dataset of Indian eye images with demographic details and hemoglobin levels. Exploratory Data Analysis (EDA) reveals insights into the dataset, showing gender, age, and hemoglobin level distributions. Scatter plots confirm that no strong correlations exist between hemoglobin levels and other features.

In the classification domain, the Decision Tree Classifier emerges as the top-performing model with an accuracy of 85.24% and an F1 score of 85.86%. Regression analysis identifies the CatBoost Regressor as the best model with an R2 score of 0.3244 and low Mean Absolute Error (MAE).

In summary, this study explores non-invasive anemia detection using anatomical images, demonstrating the potential of Decision Tree Classifier and CatBoost Regressor models for classification and regression. Future work includes data collection, augmentation, convolutional neural network utilization, further feature selection, hyperparameter tuning, and ensemble modelling to enhance accuracy and performance.

2. Preliminaries

2.1 Introduction

Hemoglobin plays a vital role in transporting oxygen within the bloodstream, which is crucial for the proper functioning of bodily organs. Alterations in hemoglobin levels, especially when they drop below the optimal range, can compromise one's health and are indicative of various medical conditions. Anemia, a condition in which the concentration of hemoglobin (Hb) falls below normal levels, fails to meet the individual's physiological requirements, affecting an estimated 1.6 billion people globally[3]. Anemia can be congenital or acquired due to chronic illnesses, but the most prevalent form is iron-deficiency anemia, often caused by inadequate dietary intake[2].

Anemia is a subtle affliction with a gradual onset, remaining asymptomatic until it reaches a severe stage. The human body compensates for the lack of oxygen, concealing symptoms. The array of symptoms associated with anemia varies, but recurring ones encompass fatigue, dizziness, headaches, pallor, chest pain, weakness, irregular heartbeats, shortness of breath, and cold extremities. In severe cases, anemia can impair cognitive and motor development in children and pose risks to pregnant women and their infants[1]. Early detection and preventive measures are crucial to mitigate severe complications and foster a healthier life.

Traditional methods for anemia detection, such as complete blood cell analysis (considered the gold standard), peripheral smears, reticulocyte counts, and serum iron indices, require invasive blood sampling[4]. However, the use of invasive methods is discouraged, especially for infants, the elderly, pregnant women, anemic patients, and those with sickle cell disease. Additionally, frequent blood sampling can be uncomfortable and expensive, particularly in economically challenged regions.

Given the limitations of invasive approaches for estimating hemoglobin levels, there is a growing demand for non-invasive methods that are user-friendly, painless, and provide rapid bedside diagnoses. Notably, as pallor is a common indicator of iron deficiency anemia[5], numerous studies have explored the use of visible pallor in exposed tissues of the body for anemia estimation. Pallor may manifest across the body but is most conspicuous in areas where blood vessels are near the surface, such as the palm, nail beds, and mucous membranes like the tongue and conjunctivae.

2.2 Related Work

Peter Appiahene *et al.*[6] utilised palmar pallor to detect anemia using various algorithms, which included CNN, k-NN, Naive Bayes, SVM, and Decision Tree. They used a primary dataset of 527 palm images to generate the region of interest (ROI) using the triangle thresholding and then augmented the size of the datasets to 2635 using the image augmentation technique (such as rotation, flipping and translation in X or Y axis) to avoid overfitting. Naïve Bayes model attained the highest accuracy of 99.96% while CNN attained the highest sensitivity of 99.98%.

Justice Williams Asare *et al.*[7] compared the conjunctiva of the eyes, the palpable palm, and the colour of the fingernail images to determine which anatomic part had a higher accuracy for detecting anemia in children. In this study, a machine learning approach was used to detect iron-deficiency anemia with the application of Naïve Bayes, CNN, SVM, k-NN, and decision tree algorithms. A Samsung Galaxy Tab 7A was utilised to capture the images from 710 children aged 6 to 59 months. Similar preprocessing to [6] was done to extract the ROI and generate the dataset for the models. After the dataset generation these models were trained, validated, and tested on images of the conjunctiva of the eyes, palpable palm, and colour of the fingernails separately. The results obtained by the models show that the CNN is robust and performs better than Naïve Bayes, decision tree, k-NN and SVM in anemia detection and the palpable palm is one of the reliable human features for anemia detection in children due to its higher detection accuracy.

Pallavi *et al.*[8] developed an Artificial Intelligence (AI)-based bot that can be used for screening people for anemia in only 35 s. The bot service is based on two models: a segmentation model to segment the Region of Interest (ROI) and a classification model to classify anemic cases from normal ones. The model got an accuracy of 97% and a sensitivity of 96% using 300 conjunctiva images and a convolution neural network.

Shaun Collings *et al.*[9] devised a screening technique for the non-invasive detection of anaemia based on digital analysis of the palpebral conjunctiva in a digital photograph. They extracted erythema index (EI) from 106 conjunctiva images and fitted a linear curve using EI as parameters, and got sensitivity of 93% and specificity of 72%

Robert G. Mannino *et al.*[10] devised a smartphone app to estimate hemoglobin level from the images of fingernail. They collected fingernail images using an Apple iPhone 5s from 337 people and extracted the ROI manually. They extracted fingernail data, skin colour data, and image metadata from ROI via MATLAB and the fitted a multi-linear regression with a bisquare weighting algorithm using skin colour data and image metadata as parameters, and achieved sensitivity of 92% and specificity of 76%

R. Mala *et al.*[11] develop a non-invasive method of anemia detection using the combined features from digital images of eye, palm, and nail bed. They extracted mean and standard deviation of a* value from CIELAB colour space and the ratio between the mean of red value to the mean of green value in RGB colour space for all the three anatomies. Using these nine features (three from each anatomy), they trained a linear regression model utilizing the images from 115 people and got accuracy, sensitivity, and specificity each as 80%.

Thomas Alan Wemyss *et al.*[12] utilised a linear combination of colorimetric metrics from the lower eyelid, conjunctiva, and the lower lip. They extracted six features namely erythema index, r chromaticity, −g chromaticity, a* chromaticity, and b* chromaticity from conjunctiva, lower eyelid, and the lower lip, and deployed a Naïve Bayes classifier to attain accuracy of 91% and sensitivity of 93%.

On conducting the literature review of 50+ papers from the past 2 decade only a very few papers (1-2) worked on the combined image parameters from all the major exposed tissues of the human body such as tongue, palm, fingernail, and conjunctiva. It was seen that each anatomic part provided a good accuracy and sensitivity, and thus combining the best parameters from each anatomy can provide a more robust and better model.

Thus, this project tries to prove the hypothesis that a combination of features from tongue, palm, fingernail, and conjunctiva (may or may not utilising all anatomies) provides a more robust machine learning model with higher accuracy and recall.

Author	Accuracy	Sensitivity	Precision	Specificity
Peter Appiahene et al [6]	0.99	0.99	0.99	NA
Justice Williams Asare et al [7]	0.98	0.99	0.99	NA
Pallavi et al [8]	0.97	0.96	0.94	NA
Shaun Collings et al [9]	NA	0.93	NA	0.72
Robert G. Mannino et al [10]	NA	0.92	NA	0.76
R. Mala et al [11]	0.8	0.8	0.94	0.8
Thomas Alan Wemyss et al [12]	0.91	0.93	0.81	0.89

Table 2.1 – Comparison of papers [19]

2.3 Machine Learning Algorithms

Before diving deep into the methodology, it is essential to understand a bit about each of most common machine learning algorithm that have been used by the researchers such as Linear Regression (LR), Support Vector Machine (SVM), Decision Tree(DT), Convolution Neural Network (CNN) to name a few.

Linear regression aims to find a linear relationship, without the need for complex mathematical equations. This technique is fundamental in understanding how one variable influence another, making it a valuable tool in various research domains. Researchers use linear regression to make predictions, identify trends, and uncover associations between variables, contributing valuable insights to a wide range of academic fields.

Support Vector Machines (SVM) is a powerful machine learning algorithm used for classification and regression tasks. Instead of relying on complex equations, SVM focuses on finding an optimal decision boundary that maximizes the separation between different classes or data points. It achieves this by identifying support vectors, which are critical data points closest to the decision boundary. SVM's ability to handle high-dimensional data and non-linear relationships makes it a popular choice in various research and analysis scenarios, offering robust solutions to classification problems.

Decision trees are intuitive and interpretable machine learning models. They resemble hierarchical flowcharts where each internal node represents a decision based on a specific feature, leading to subsequent nodes or leaves that represent outcomes or predictions. By recursively splitting data, decision trees make choices that maximize information gain or minimize impurity, allowing them to efficiently partition datasets into classes or values. These versatile structures are widely used for decision-making processes across diverse fields, offering a straightforward yet effective approach to problem-solving and analysis.

Convolutional Neural Networks (CNNs) are specialized deep learning models designed to process and analyse visual data, like images and videos. Unlike traditional neural networks, CNNs use convolutional layers that apply filters to capture patterns and features hierarchically. These filters learn to detect various aspects such as edges, textures, and shapes within the data. By doing so, CNNs enable effective feature extraction, making them indispensable in computer vision tasks. CNNs' architecture, inspired by the visual processing in the human brain, has revolutionized image analysis and recognition.

3. Methodology

The general idea of methodology is shown in fig 3.1. The process begins with the segmentation of conjunctiva from the eye images to extract the region of interest (ROI), then data augmentation such as rotation along the axis, enlargement, flipping etc. is done to make the model robust to the dataset. After this feature extraction is performed independently in the RGB space, the HSV space and the Gray space. Now, exploratory data analysis (EDA) is done on the features to understand the nature of algorithm that can capture the relationship between the dependent and the independent variables. Using the insights from the EDA, feature selection is carried out to check for highly correlated features. Now the dataset is split into training and testing, and different models are fitted on the training dataset, after which their performances are evaluated using the unseen (test) dataset through different metrics depending on whether classification is being performed or regression analysis.

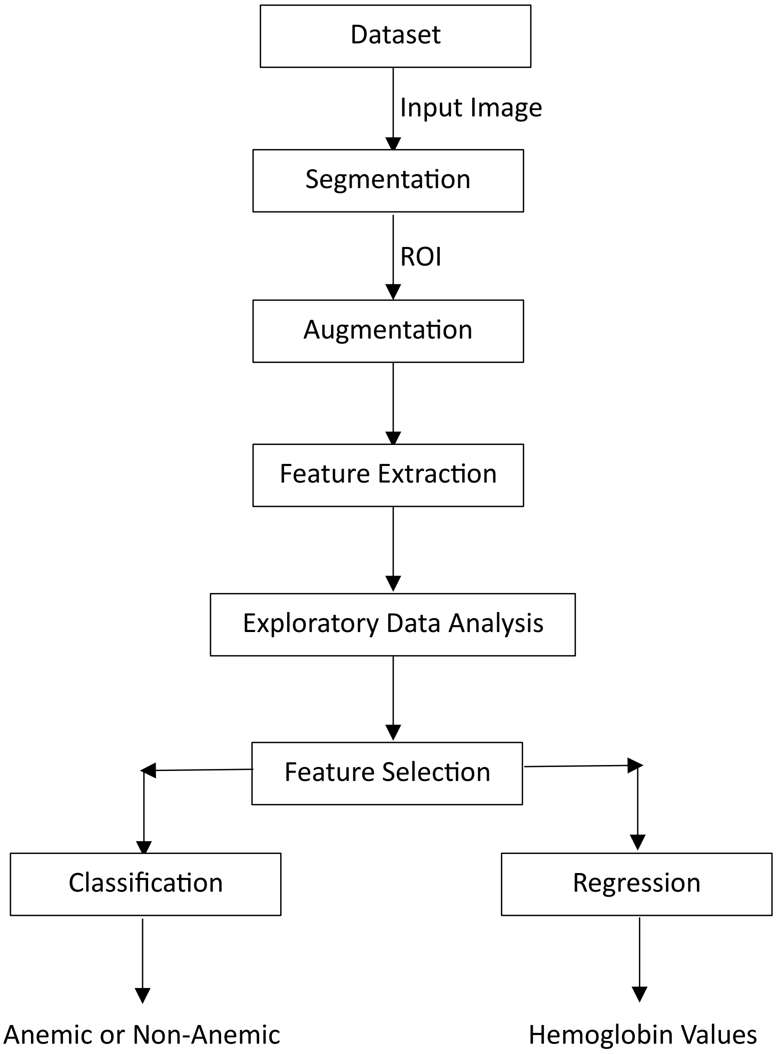
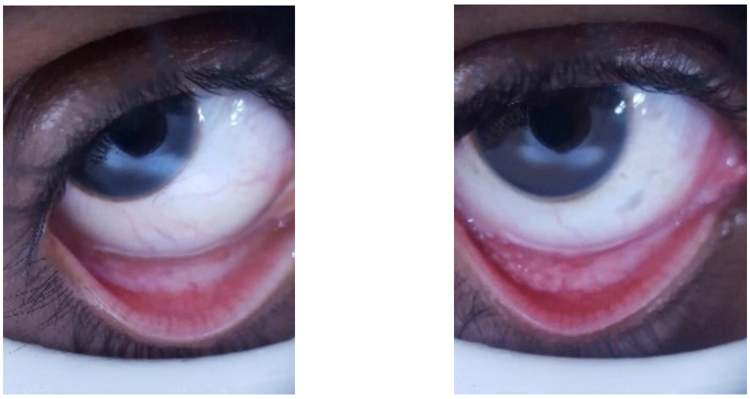


Fig. 3.1 – Flowchart of methodology

3.1 Dataset

The dataset that has been used for this part of the project contains 95 Indian eye images[13] of conjunctiva along with the segmented region of interest i.e., forniceal conjunctiva, palpebral conjunctiva, both, the patients' demographics such as age and gender, and finally the hemoglobin level when the photo of eye was taken.



a) Anemic b) Non-Anemic

Fig. 3.2 – Shows eye images from the dataset [13]

3.2 Segmentation

For this part of project, since the ROI of conjunctiva was already present, no segmentation was done. But in case of non-availability of ROI, the region of interest can be extracted using Sabrina's method[14] i.e., using deep neural architectures such as UNet, FCN, PSPNet, or LinkNet.

a) Anemic b) Non-Anemic

Fig. 3.3 – Shows segmented conjunctiva [13]

3.3 Data Augmentation

No data augmentation was required up until now in the project. But augmentations such as rotation, cropping, adding noise, flipping, scaling, contrast, translation etc, can be done using different libraries as OpenCV[15], Imgaug[16], etc in python.

3.4 Feature Extraction

A total of twelve features were extracted as mentioned in [17] these are $mean_r$, $mean_g$, $mean_b$, $mean_{r-g}$, HHR, Ent, B, g1, g2, g3, g4, and g5 were extracted from each of the ROI of conjunctiva.

Features $mean_r$, $mean_g$, $mean_b$, $mean_{r-g}$ were extracted from the RGB colour space such that these features are mean intensity values of the red, green, and blue components of the ROI and $mean_{r-g} = mean(r_i-g_i)$

HHR is called the high hue ratio, and it represents the proportion of pixels with high values in the hue component of the image in HSV space. It is given by the ratio of number of high-hue pixels in an ROI, with a total of N pixels. A high HHR value denotes a large red area in an image.

From the grey space features like Entropy (Ent), average brightness (B) and the g(1 to 5) features are extracted. The formulae can be found in section 3.2 (Feature Extraction) in [17]

Along with these 12 features, age and gender were included in the final list of features, thus making a total of 14 independent variables, with label/hemoglobin as dependent variable.

See Annexure A1 for the code of feature extraction.

3.5 Exploratory Data Analysis

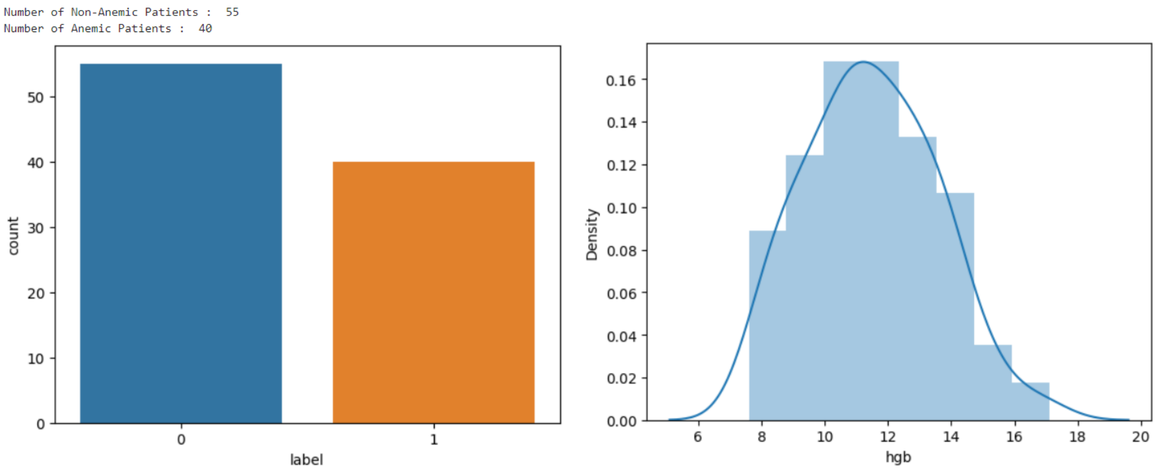
Exploratory Data Analysis (EDA) entails the systematic examination of data to uncover patterns, relationships, and anomalies. Techniques like summary statistics, data visualization, and data cleaning are employed to gain insights and identify data quality issues. EDA not only enhances data comprehension but also helps in making informed decisions on subsequent analytical approaches.

3.5.1 Hemoglobin (Hgb) levels

According to WHO, the hemoglobin level for anemia varies according to sex, age and whether a woman is pregnant or not. For the sake of simplicity, I have combined the anemic and moderately anemic across all age and gender as “anemic” with label as **1** and rest non-anemic and mildly anemic as “non-anemic” with label as **0**, with the threshold as **11.0** g/dL [18].

Fig. 3.4(left) shows the distribution of labels in the dataset. There are 55 non-anemic patients with label **0** while there are 40 anemic patients with label **1**

Fig. 3.4(right) shows the histogram of hemoglobin (hgb) levels in the patients. The highest bars are for the bucket 10-12 and 12-14 hgb signifying that most of the patients in the dataset have hemoglobin between 10-14 g/dL



```
Number of Non-Anemic Patients :  55
Number of Anemic Patients :  40
```

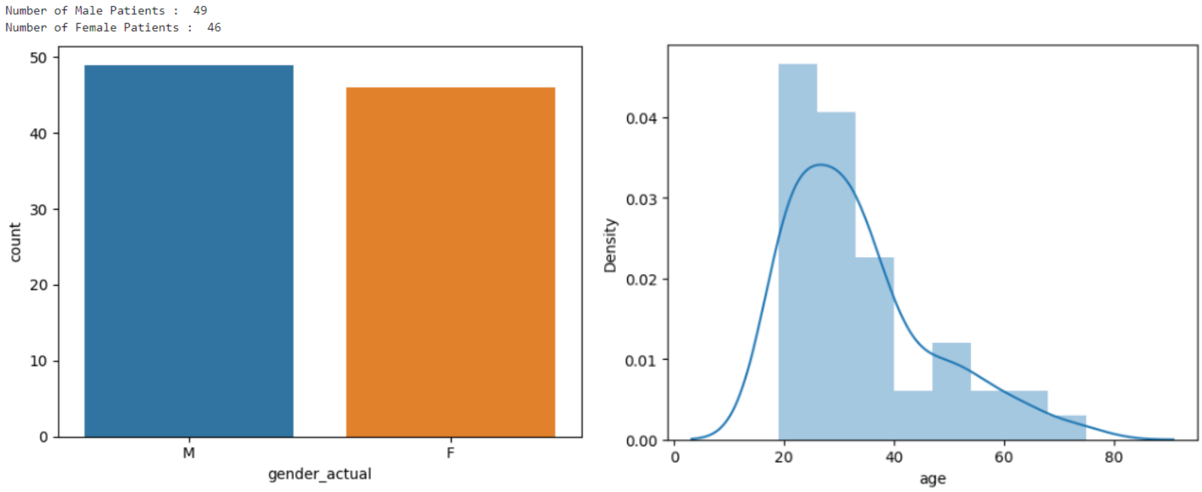
Fig. 3.4 - Bar chart of labels and Histogram of hemoglobin level. Left: A bar chart representing the number of labels. Right: A histogram representing the distribution of hgb levels

3.5.2 Age and Gender

Other than hemoglobin, age and gender data were also collected from the patients.

Fig. 3.5(left) shows the distribution of gender in the dataset. There are 49 male patients with label **M** while there are 46 female patients with label **F.**

Fig. 3.5(right) shows the histogram of age in the patients. The highest bar is for the bucket 20-25 years signifying that most of the patients in the dataset have age between 20-25 years.



```
Number of Male Patients :  49
Number of Female Patients :  46
```

Fig. 3.5 – Bar chart of gender and Histogram of age. Left: A bar chart representing the number of males and females in the dataset. Right: A histogram representing the distribution of age

3.5.3 Scatter Plot

A scatter plot presents a graphical representation of data points on a two-dimensional plane, with one variable on the x-axis and another on the y-axis. This visualization technique helps in exploring the relationship between two variables, making it invaluable in hypothesis testing and data analysis. Scatter plots can reveal trends, correlations, outliers, and clusters within the data, aiding in the identification of patterns and potential insights.

Fig 3.6 – represents the scatter plot of hemoglobin level vs other features in the dataset. In all the graphs it is evident that there is general trend in points placement. Thus, this means that there is no correlation between hemoglobin and the following features.

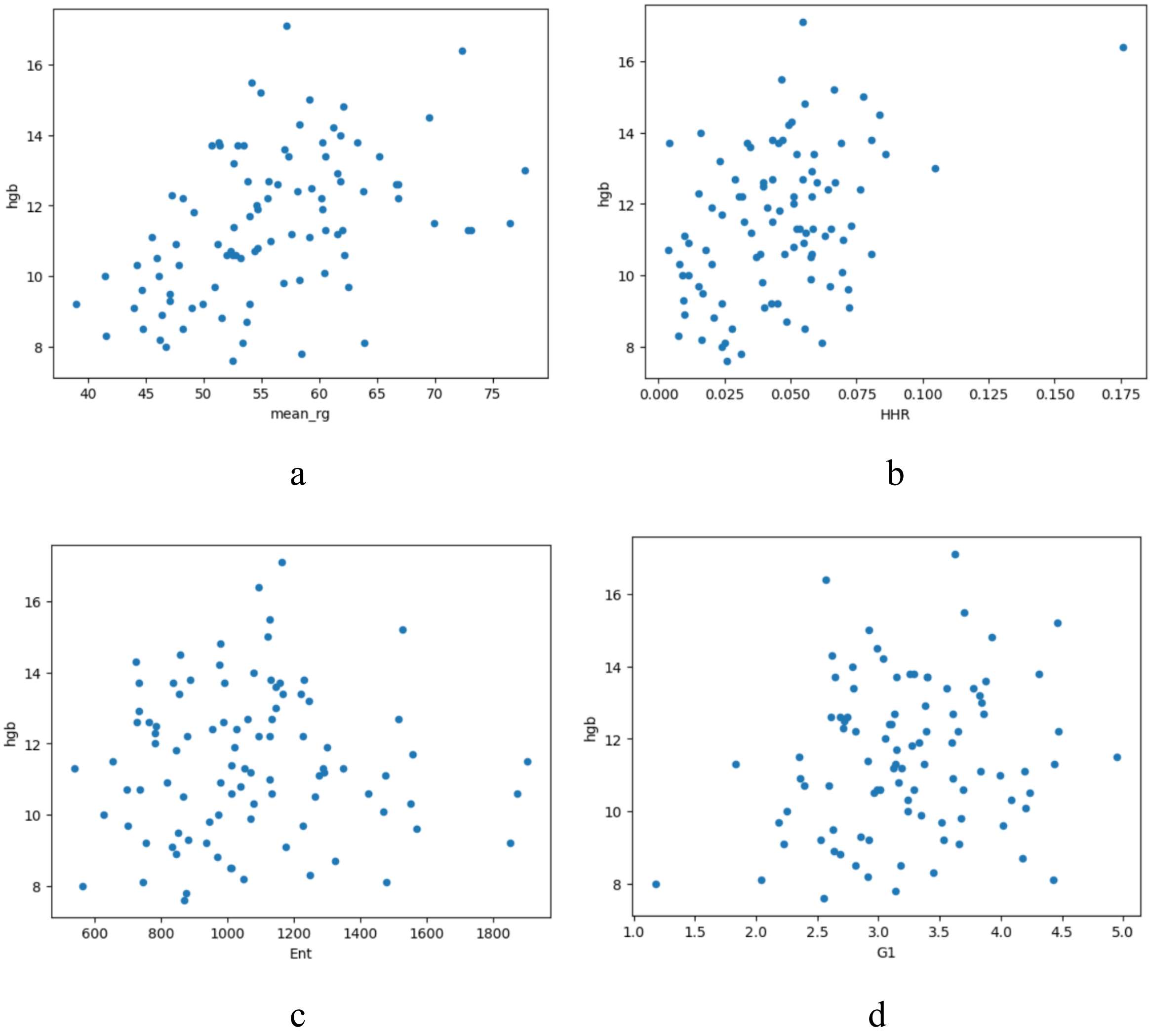


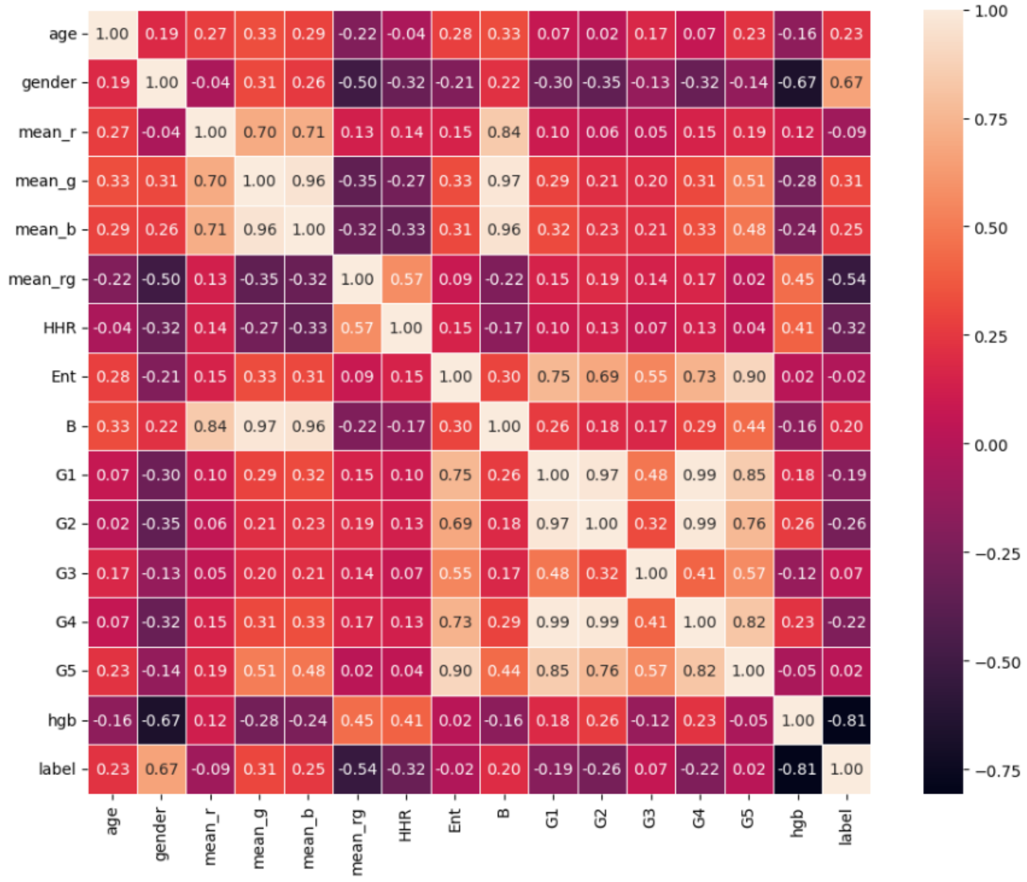
Fig. 3.6 – Scatter plot between hgb and other features in the dataset

a: hgb vs mean_rg, b: hgb vs HHR, c: hgb vs Ent, d: hgb vs G1

3.6 Feature Selection

Feature selection involves the systematic identification of the most consistent, non-redundant, and relevant features to be employed in model development. This process is of utmost significance as datasets continue to expand in size and diversity. The primary objective of feature selection is twofold: to enhance the predictive model's performance and to curtail the computational overhead of modelling.

Correlation serves as a metric for quantifying the linear association between two or more variables, enabling us to anticipate one variable based on another. The rationale behind employing correlation for feature selection lies in the notion that valuable variables exhibit a strong correlation with the target variable. Additionally, these variables should exhibit a correlation with the target variable while remaining uncorrelated with each other. When two variables exhibit a strong correlation, one can be predicted from the other. Consequently, in cases of feature correlation, the model only requires one of the correlated features, as the second does not contribute any additional information.



	age	gender	mean_r	mean_g	mean_b	mean_rg	HHR	Ent	B	G1	G2	G3	G4	G5	hgb	label
age	1.00	0.19	0.27	0.33	0.29	-0.22	-0.04	0.28	0.33	0.07	0.02	0.17	0.07	0.23	-0.16	0.23
gender	0.19	1.00	-0.04	0.31	0.26	-0.50	-0.32	-0.21	0.22	-0.30	-0.35	-0.13	-0.32	-0.14	-0.67	0.67
mean_r	0.27	-0.04	1.00	0.70	0.71	0.13	0.14	0.15	0.84	0.10	0.06	0.05	0.15	0.19	0.12	-0.09
mean_g	0.33	0.31	0.70	1.00	0.96	-0.35	-0.27	0.33	0.97	0.29	0.21	0.20	0.31	0.51	-0.28	0.31
mean_b	0.29	0.26	0.71	0.96	1.00	-0.32	-0.33	0.31	0.96	0.32	0.23	0.21	0.33	0.48	-0.24	0.25
mean_rg	-0.22	-0.50	0.13	-0.35	-0.32	1.00	0.57	0.09	-0.22	0.15	0.19	0.14	0.17	0.02	0.45	-0.54
HHR	-0.04	-0.32	0.14	-0.27	-0.33	0.57	1.00	0.15	-0.17	0.10	0.13	0.07	0.13	0.04	0.41	-0.32
Ent	0.28	-0.21	0.15	0.33	0.31	0.09	0.15	1.00	0.30	0.75	0.69	0.55	0.73	0.90	0.02	-0.02
B	0.33	0.22	0.84	0.97	0.96	-0.22	-0.17	0.30	1.00	0.26	0.18	0.17	0.29	0.44	-0.16	0.20
G1	0.07	-0.30	0.10	0.29	0.32	0.15	0.10	0.75	0.26	1.00	0.97	0.48	0.99	0.85	0.18	-0.19
G2	0.02	-0.35	0.06	0.21	0.23	0.19	0.13	0.69	0.18	0.97	1.00	0.32	0.99	0.76	0.26	-0.26
G3	0.17	-0.13	0.05	0.20	0.21	0.14	0.07	0.55	0.17	0.48	0.32	1.00	0.41	0.57	-0.12	0.07
G4	0.07	-0.32	0.15	0.31	0.33	0.17	0.13	0.73	0.29	0.99	0.99	0.41	1.00	0.82	0.23	-0.22
G5	0.23	-0.14	0.19	0.51	0.48	0.02	0.04	0.90	0.44	0.85	0.76	0.57	0.82	1.00	-0.05	0.02
hgb	-0.16	-0.67	0.12	-0.28	-0.24	0.45	0.41	0.02	-0.16	0.18	0.26	-0.12	0.23	-0.05	1.00	-0.81
label	0.23	0.67	-0.09	0.31	0.25	-0.54	-0.32	-0.02	0.20	-0.19	-0.26	0.07	-0.22	0.02	-0.81	1.00

Fig. 3.7 – A heatmap of correlation between all the features and independent variable

4. Results

This project can be accomplished using classification (classifying as anemic or non-anemic) as well as regression (predicting the hemoglobin value). From a medical point of view the hemoglobin prediction is much more general case and predicting anemia is just a consequence of this. For this stage, classification has been more focused to replicate the results achieved in the literature review. But a regression analysis had also been done to understand the feature importance and evaluation metric and compare it with the classification task.

To develop and compare models PyCaret[20] library has been used. PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows.

4.1 Classification

4.1.1 Evaluation Metrics

To evaluate the performance of the classification algorithms, metrics such as accuracy, AUC, Recall, F1 etc has been used. The images correctly categorized by the classifier as part of the anemic class are denoted as true positives (TP), while those accurately identified as not anemic are referred to as true negatives (TN). The images of non-anemic patients that were mistakenly classified as anemic are categorized as false positives (FPs). Lastly, the conjunctival images of anemic patients that were erroneously classified as non-anemic belong to the category of false negatives (FNs). Based on these notations, accuracy, sensitivity, and specificity are defined as follows:

$$\text{accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{recall or sensitivity} = TP / (TP + FN)$$

$$\text{specificity} = TN / (TN + FP)$$

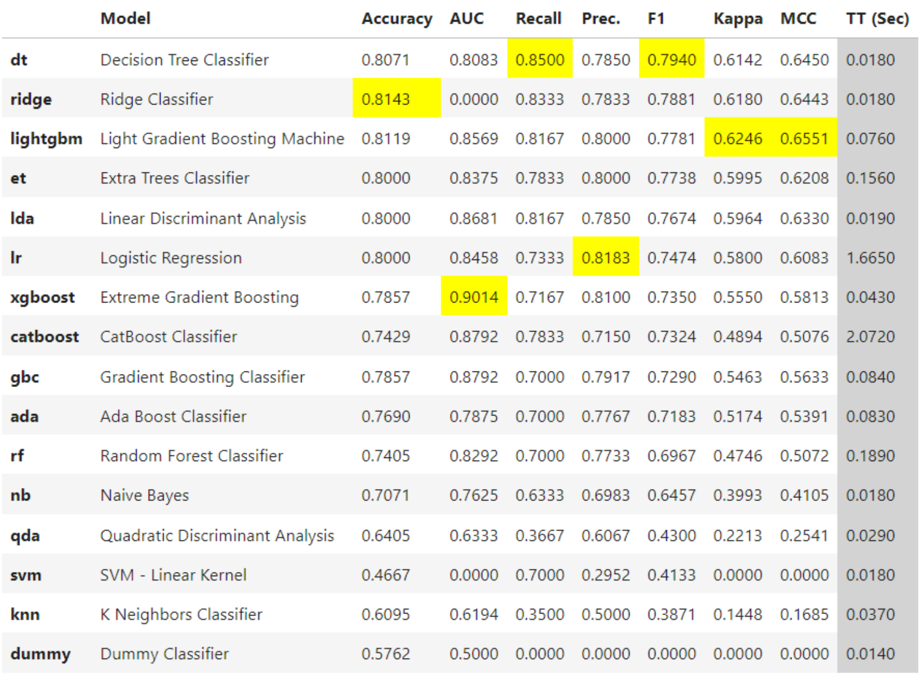
$$\text{precision} = TP / (TP + FP)$$

$$\text{F1 score} = 2*(\text{precision}*\text{recall}) / (\text{precision} + \text{recall})$$

4.1.2 Model Comparison

Fig 4.1 shows the comparison of all the classification models using the PyCaret classification library's compare_model function. All the models are tested and ranked according to their recall (or sensitivity) score. It is evident that Decision Tree(dt) Classifier has the highest recall and the highest F1 score.

See Annexure A2 for the python code for classification



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	0.8071	0.8083	0.8500	0.7850	0.7940	0.6142	0.6450	0.0180
ridge	Ridge Classifier	0.8143	0.0000	0.8333	0.7833	0.7881	0.6180	0.6443	0.0180
lightgbm	Light Gradient Boosting Machine	0.8119	0.8569	0.8167	0.8000	0.7781	0.6246	0.6551	0.0760
et	Extra Trees Classifier	0.8000	0.8375	0.7833	0.8000	0.7738	0.5995	0.6208	0.1560
lda	Linear Discriminant Analysis	0.8000	0.8681	0.8167	0.7850	0.7674	0.5964	0.6330	0.0190
lr	Logistic Regression	0.8000	0.8458	0.7333	0.8183	0.7474	0.5800	0.6083	1.6650
xgboost	Extreme Gradient Boosting	0.7857	0.9014	0.7167	0.8100	0.7350	0.5550	0.5813	0.0430
catboost	CatBoost Classifier	0.7429	0.8792	0.7833	0.7150	0.7324	0.4894	0.5076	2.0720
gbc	Gradient Boosting Classifier	0.7857	0.8792	0.7000	0.7917	0.7290	0.5463	0.5633	0.0840
ada	Ada Boost Classifier	0.7690	0.7875	0.7000	0.7767	0.7183	0.5174	0.5391	0.0830
rf	Random Forest Classifier	0.7405	0.8292	0.7000	0.7733	0.6967	0.4746	0.5072	0.1890
nb	Naive Bayes	0.7071	0.7625	0.6333	0.6983	0.6457	0.3993	0.4105	0.0180
qda	Quadratic Discriminant Analysis	0.6405	0.6333	0.3667	0.6067	0.4300	0.2213	0.2541	0.0290
svm	SVM - Linear Kernel	0.4667	0.0000	0.7000	0.2952	0.4133	0.0000	0.0000	0.0180
knn	K Neighbors Classifier	0.6095	0.6194	0.3500	0.5000	0.3871	0.1448	0.1685	0.0370
dummy	Dummy Classifier	0.5762	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0140

Fig. 4.1 – Comparison of all classification models

The confusion matrix provides a concise overview of a machine learning model's performance when tested on a test dataset. It serves as a valuable tool for evaluating the effectiveness of classification models, which are designed to predict a categorical label for each given input instance. Fig 4.2 shows the confusion matrix for the decision tree classifier. For the given confusion matrix TP=17,TN=5,FP=0;FN=7; this implies Accuracy of 75.86%, Recall of 41.67%, Precision of 100.00% and F1 score of 58.82%.

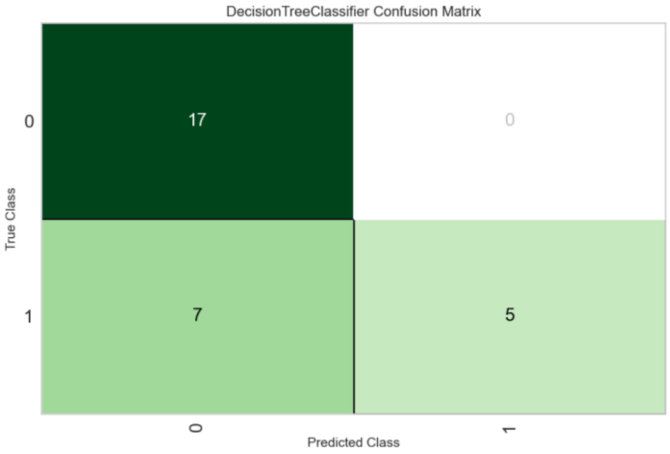
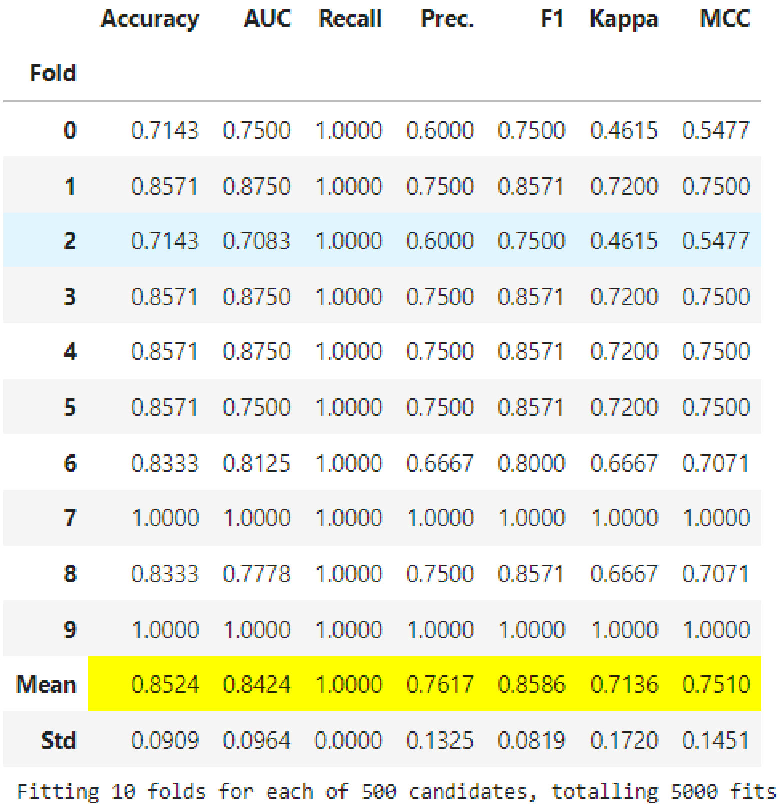


Fig. 4.2 – Confusion Matrix for Decision Tree Classifier

4.1.3 Tuning

From the confusion matrix it is evident that the model has a poor recall, though it has the highest precision and good accuracy. Thus, the model needs some tuning to improve the metrics and achieve higher scores.
Fig 4.3 shows the summary of hyperparameter tuning for decision tree model



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	0.7143	0.7500	1.0000	0.6000	0.7500	0.4615	0.5477
1	0.8571	0.8750	1.0000	0.7500	0.8571	0.7200	0.7500
2	0.7143	0.7083	1.0000	0.6000	0.7500	0.4615	0.5477
3	0.8571	0.8750	1.0000	0.7500	0.8571	0.7200	0.7500
4	0.8571	0.8750	1.0000	0.7500	0.8571	0.7200	0.7500
5	0.8571	0.7500	1.0000	0.7500	0.8571	0.7200	0.7500
6	0.8333	0.8125	1.0000	0.6667	0.8000	0.6667	0.7071
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	0.8333	0.7778	1.0000	0.7500	0.8571	0.6667	0.7071
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Mean	0.8524	0.8424	1.0000	0.7617	0.8586	0.7136	0.7510
Std	0.0909	0.0964	0.0000	0.1325	0.0819	0.1720	0.1451

```
Fitting 10 folds for each of 500 candidates, totalling 5000 fits
```

Fig. 4.3 – Summary of tuning the dt model

4.2 Regression

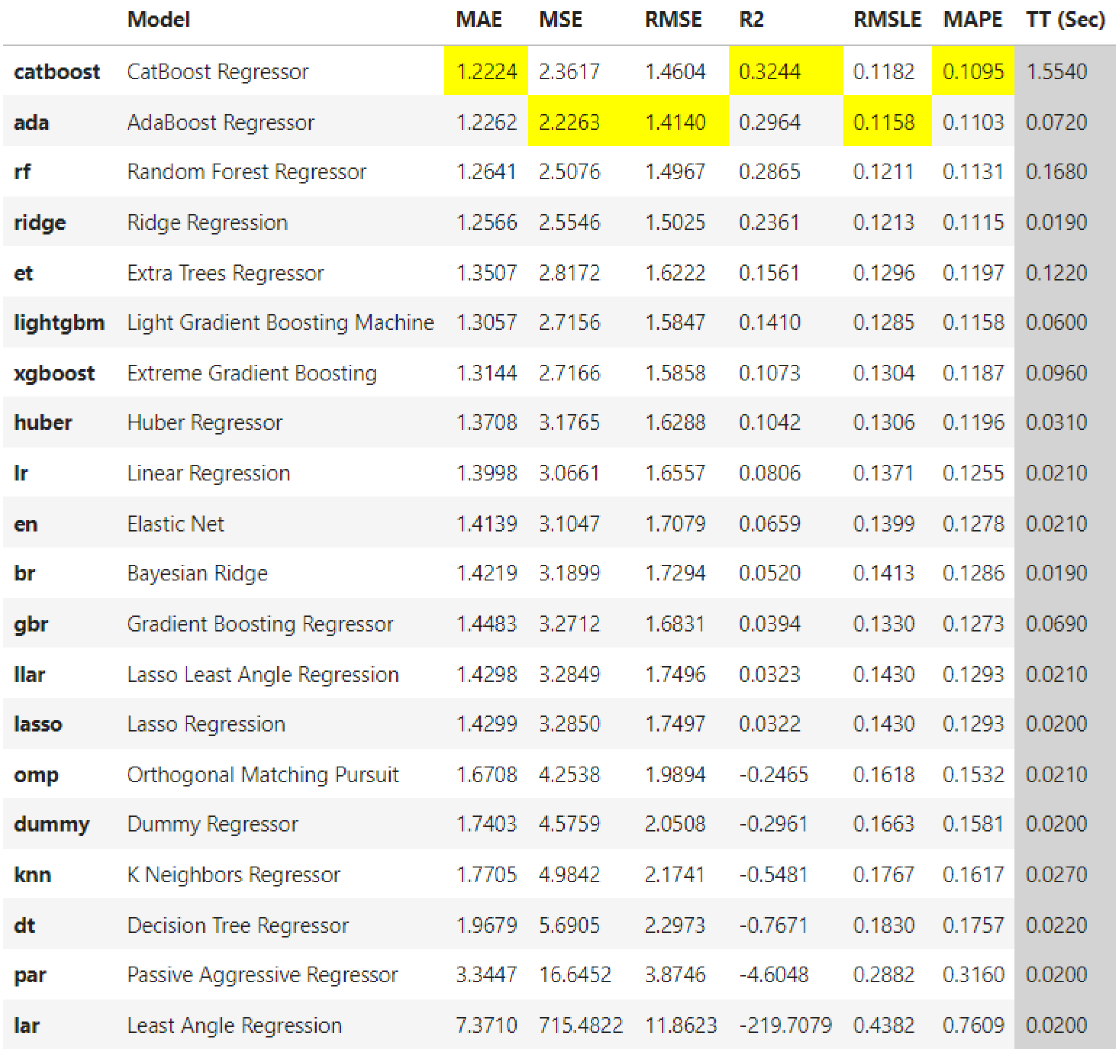
4.2.1 Evaluation Metrics

In assessing the performance of regression algorithms, various metrics are employed, including mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-squared (R2). MAE quantifies the absolute disparity between actual and predicted values. Mean squared error (MSE), alternatively known as mean squared deviation (MSD), computes the average of the squared discrepancies between observed values in a statistical study and those predicted by a model. RMSE, on the other hand, is the square root of the mean squared error. The R2 score pertains to the model's performance and not the loss in an absolute context. It gauges how much the regression line outperforms a mean line. Hence, R2 squared is also referred to as the Coefficient of Determination or, at times, the Goodness of Fit.

4.2.2 Model Comparison

Fig 4.4 shows the comparison of all the regression models using the PyCaret regression library's compare_model function. All the models are tested and ranked according to their R2 score. It is evident that CatBoost Classifier has the highest R2 score and the lowest MAE score.

See Annexure A3 for the python code for regression



	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	1.2224	2.3617	1.4604	0.3244	0.1182	0.1095	1.5540
ada	AdaBoost Regressor	1.2262	2.2263	1.4140	0.2964	0.1158	0.1103	0.0720
rf	Random Forest Regressor	1.2641	2.5076	1.4967	0.2865	0.1211	0.1131	0.1680
ridge	Ridge Regression	1.2566	2.5546	1.5025	0.2361	0.1213	0.1115	0.0190
et	Extra Trees Regressor	1.3507	2.8172	1.6222	0.1561	0.1296	0.1197	0.1220
lightgbm	Light Gradient Boosting Machine	1.3057	2.7156	1.5847	0.1410	0.1285	0.1158	0.0600
xgboost	Extreme Gradient Boosting	1.3144	2.7166	1.5858	0.1073	0.1304	0.1187	0.0960
huber	Huber Regressor	1.3708	3.1765	1.6288	0.1042	0.1306	0.1196	0.0310
lr	Linear Regression	1.3998	3.0661	1.6557	0.0806	0.1371	0.1255	0.0210
en	Elastic Net	1.4139	3.1047	1.7079	0.0659	0.1399	0.1278	0.0210
br	Bayesian Ridge	1.4219	3.1899	1.7294	0.0520	0.1413	0.1286	0.0190
gbr	Gradient Boosting Regressor	1.4483	3.2712	1.6831	0.0394	0.1330	0.1273	0.0690
llar	Lasso Least Angle Regression	1.4298	3.2849	1.7496	0.0323	0.1430	0.1293	0.0210
lasso	Lasso Regression	1.4299	3.2850	1.7497	0.0322	0.1430	0.1293	0.0200
omp	Orthogonal Matching Pursuit	1.6708	4.2538	1.9894	-0.2465	0.1618	0.1532	0.0210
dummy	Dummy Regressor	1.7403	4.5759	2.0508	-0.2961	0.1663	0.1581	0.0200
knn	K Neighbors Regressor	1.7705	4.9842	2.1741	-0.5481	0.1767	0.1617	0.0270
dt	Decision Tree Regressor	1.9679	5.6905	2.2973	-0.7671	0.1830	0.1757	0.0220
par	Passive Aggressive Regressor	3.3447	16.6452	3.8746	-4.6048	0.2882	0.3160	0.0200
lar	Least Angle Regression	7.3710	715.4822	11.8623	-219.7079	0.4382	0.7609	0.0200

Fig. 4.4 – Comparison of all regression models

Fig 4.5 shows the residuals vs. fits plot. Points to note from the plot are

- The train R2=1 and test R2=0.454. This suggests that the model can perfectly explain the relationship between the dependent and independent variables in training but only 45.4% of the relationship during testing indicating high variance in the model

- No one residual "stands out" from the basic random pattern of residuals. This suggests that there are no outliers.

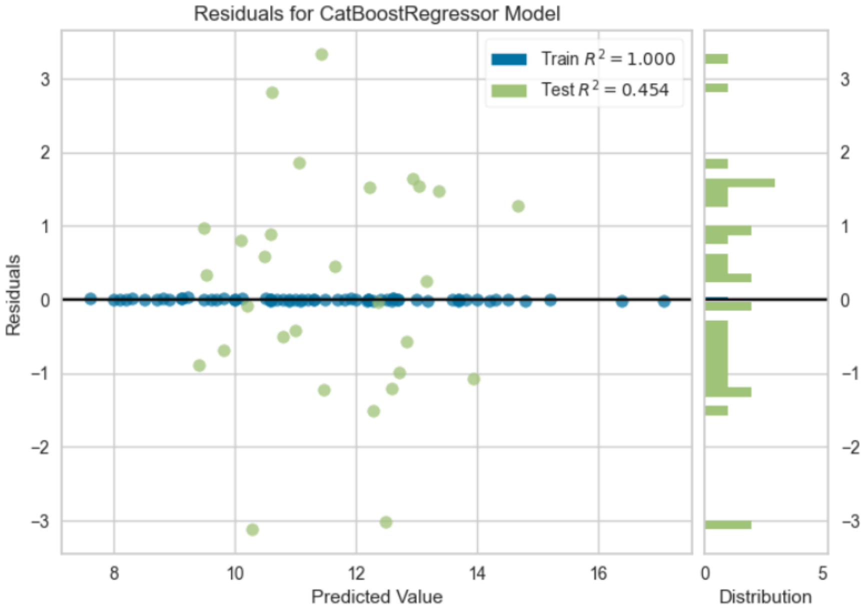
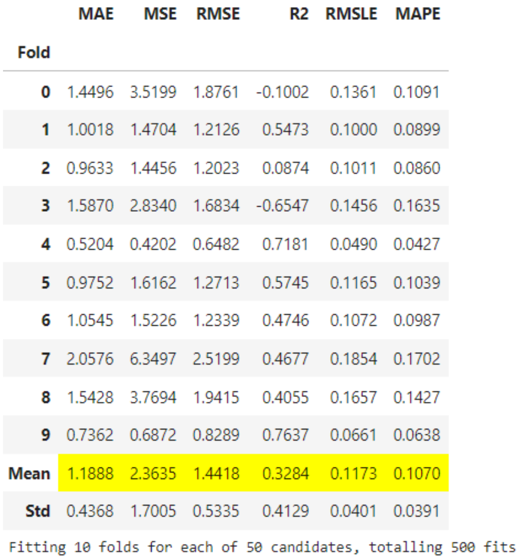


Fig. 4.5 – Residual vs Predicted Value Plot

4.2.3 Tuning

From the fig 4.4 it is evident that the model still has some error in MAE or RMSE and R2 score is 0.3244, this suggests that model explains 32% of the relationship between the dependent and independent variables. Thus Fig 4.6 shows tuning of the CatBoost model



Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	1.4496	3.5199	1.8761	-0.1002	0.1361	0.1091
1	1.0018	1.4704	1.2126	0.5473	0.1000	0.0899
2	0.9633	1.4456	1.2023	0.0874	0.1011	0.0860
3	1.5870	2.8340	1.6834	-0.6547	0.1456	0.1635
4	0.5204	0.4202	0.6482	0.7181	0.0490	0.0427
5	0.9752	1.6162	1.2713	0.5745	0.1165	0.1039
6	1.0545	1.5226	1.2339	0.4746	0.1072	0.0987
7	2.0576	6.3497	2.5199	0.4677	0.1854	0.1702
8	1.5428	3.7694	1.9415	0.4055	0.1657	0.1427
9	0.7362	0.6872	0.8289	0.7637	0.0661	0.0638
Mean	1.1888	2.3635	1.4418	0.3284	0.1173	0.1070
Std	0.4368	1.7005	0.5335	0.4129	0.0401	0.0391

```
Fitting 10 folds for each of 50 candidates, totalling 500 fits
```

Fig. 4.6 – Summary of tuning the CatBoost model

4.3 Comparing Classification and Regression Techniques

From fig 4.7 it is evident that gender and HHR(high hue ratio) are two most prominent features for both classification and regression.

For classification, the feature importance falls drastically after the 6th feature.

For regression, there is a huge difference between the importance of gender and HHR but after HHR all the features are almost equally important.

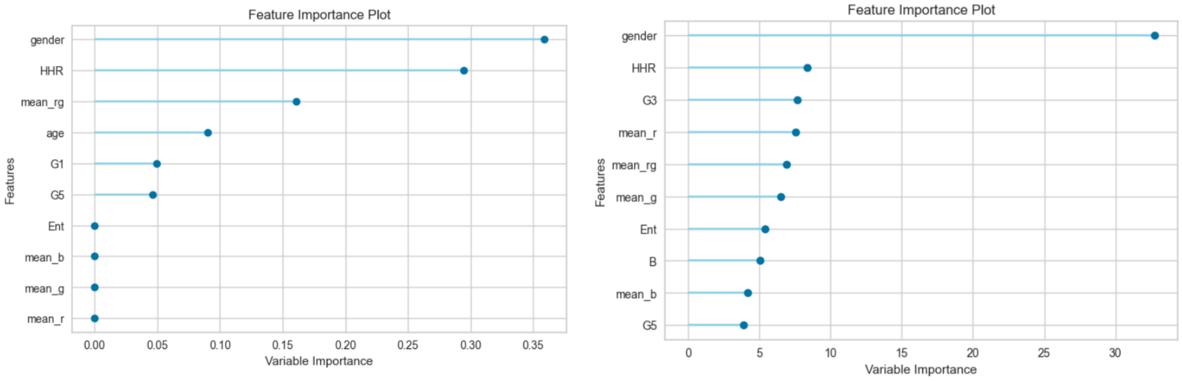


Fig. 4.7 – Feature Importance Plot. Left: for classification. Right: for regression

5. Summary and Future Work

5.1 Summary

- Anaemia has a slow evolution, and it does not manifest symptoms until it becomes severe
- Close to 1.6B people suffer from anemia globally
- In the past two decades there has been a shift to estimate hemoglobin non-invasively, and in the past 3-4 years there has been a rise in using anatomic images (such as eye, nail, palm etc) to detect anemia
- General methodology followed is to first segment the image and extract the ROI and then perform feature extraction followed by training and testing
- Best Classification model so far is Decision Tree with accuracy of 85.24% and F1 score of 85.86%
- Six most important feature for classification (in order of importance) were gender, HHR, mean_rg, age, G1 and G5
- Best Regression model so far is CatBoost Regressor with R2 of 0.3244 and MAE of 1.224
- Six most important feature for regression (in order of importance) were gender, HHR, G3, mean_r, mean_rg and mean_g
- To increase the performance of the models, data augmentation, feature selection and more hyperparameter tuning could be tried out

5.2 Future Work

5.2.1 Collection of Data

- Design a research protocol for data collection
- Collect additional data such as fingernail, palm, tongue images along with the hemoglobin levels and demographics such as age, gender etc
- Devise a strategy to store and clean the data for future use

5.2.2 Existing Data

- Perform data augmentation to increase the dataset
- Try out the convolutional neural network for both classification and regression
- Perform feature selection or PCA to reduce dimensionality of the features
- Perform hyperparameter tuning for all the top 3 models with the best metrics in classification as well as regression
- Try out ensemble models by combining the predictions from the weaker models

6. Annexure

A1 Code for feature extraction

```
""" Images should be sent in RGB format """
def feature(data):
    """Return all the 12 features as a numpy array"""
    number,img,label = data
    img = resize(img,(250,250))

    #RGB SPACE
    r, g, b = cv2.split(img)
    sum_img = [sum(r),sum(g),sum(b),sum(r-g)]
    mean_features = [i[0]/i[1] for i in sum_img]
    mean_r,mean_g,mean_b,mean_rg = mean_features
    # 4 features done in RGB SPACE


    #HSV SPACE
    hsv = to_hsv(img)
    h,s,v = cv2.split(hsv)
    h = h/h.max()
    nH = np.count_nonzero(h>0.95)
    HHR = nH/h.size
    # HHR found


    #GRAY SPACE
    gray = to_gray(img)
    B_sum, B_size = sum(gray)
    B = B_sum/B_size # FOUND B

    #ENTROPY in gray space
    eq = cv2.equalizeHist(gray)
    unique, counts = np.unique(eq, return_counts=True)
    #only pixels whose value is between 20 and 240
    total_counts = counts[21:240].sum()
    Ent = np.sum(np.array([-i*(i/total_counts)*math.log((i/total_counts),2) for i in counts[21:240]])) #Found Entropy

    #Calculating the 'G' features
    Ixy = gray
    min_Ixy = pooling(image=Ixy, pool_size=(3,3), code='min', padding=1)
    max_Ixy = pooling(image=Ixy, pool_size=(3,3), code='max', padding=1)
    mean_Ixy = pooling(image=Ixy, pool_size=(3,3), code='mean', padding=1)
    std_Ixy = pooling(image=Ixy, pool_size=(3,3), code='std', padding=1)

    g1 = Ixy - min_Ixy
    g2 = max_Ixy - Ixy
    g3 = Ixy - mean_Ixy
    g4 = std_Ixy
    g5 = Ixy

    G1 = g1.sum()/g1.size
    G2 = g2.sum()/g2.size
    G3 = g3.sum()/g3.size
    G4 = g4.sum()/g4.size
    G5 = g5.sum()/g5.size

    feature_all = [number, mean_r, mean_g, mean_b, mean_rg, HHR, Ent, B, G1, G2, G3, G4, G5, label]
    return feature_all
```

Points to Note in the above function:

- Images sent to the above function are in format number (name of image), image (in RGB format), label
- resize() – It returns the resized image
- sum() – This function takes input 1 component of an image (i.e., either red, blue, green, or gray) and returns the (sum,count), where sum represents the sum of pixels having pixel values between 20 and 240 while count represents the count of such pixels
- to_hsv() / to_gray() – These functions convert the image to HSV and Gray space respectively
- pooling() – This function returns a pooled image with padding

A2 Code for Classification

```
from pycaret.classification import *

s = setup(data, target='label', ignore_features=['hgb','number'], preprocess=False, session_id=123)

best_c = compare_models(sort = 'F1')

evaluate_model(best_c)

predict_model(best_c)

tuned_dt = tune_model(best_c, fold = 10, n_iter = 500, choose_better = True)
```

A3 Code for Regression

```
from pycaret.regression import *

s = setup(data, target='hgb', ignore_features=['label','number'], preprocess=False, session_id=123)

best_r = compare_models()

evaluate_model(best_r)

predict_model(best_r)

tuned_catboost = tune_model(best_r, fold = 10, n_iter = 50)
```

Points to Note[21]:

- setup() - initializes the training environment and creates the transformation pipeline
- compare_models() - trains and evaluates performance of all estimators available in the model library using cross validation
- evaluate_model() - displays a user interface for analysing performance of a trained model
- predict_model() - predicts Label and Score (probability of predicted class) using a trained model
- tune_model() - tunes the hyperparameters of a given estimator

7. References

1. Anemia – World Health Organisation https://www.who.int/anaemia
2. Anemia https://my.clevelandclinic.org/health/diseases/3929-anemia
3. McLean, E., Cogswell, M., Egli, I., Wojdyla, D., & De Benoist, B. (2009). Worldwide prevalence of anaemia, WHO Vitamin and Mineral Nutrition Information System, 1993–2005. Public Health Nutrition, 12(4), 444-454. doi:10.1017/S1368980008002401
4. Johnson-Wimbley TD, Graham DY. Diagnosis and management of iron deficiency anemia in the 21st century. Therapeutic Advances in Gastroenterology. 2011;4(3):177-184. doi:10.1177/1756283X11398736
5. Iron deficiency anemia mayoclinic.org/iron-deficiency-anemia
6. Appiahene, P., Asare, J.W., Donkoh, E.T. et al. Detection of iron deficiency anemia by medical images: a comparative study of machine learning algorithms. BioData Mining 16, 2 (2023). doi:10.1186/s13040-023-00319-z
7. Asare, JW, Appiahene, P, Donkoh, ET, Dimauro, G. Iron deficiency anemia detection using machine learning models: A comparative study of fingernails, palm and conjunctiva of the eye images. Engineering Reports. 2023;e12667. doi:10.1002/eng2.12667
8. Pallavi, Bijit Basumatary, Rahul Shukla, Rakesh Kumar, Bodhisatwa Das & Ashish Kumar Sahani (2023) A Deep Learning-based System for Detecting Anemia from Eye Conjunctiva Images Taken from a Smartphone, IETE Technical Review, DOI: 10.1080/02564602.2023.2242318
9. Collings S, Thompson O, Hirst E, Goossens L, George A, et al. (2016) Non-Invasive Detection of Anaemia Using Digital Photographs of the Conjunctiva. PLOS ONE 11(4): e0153286. https://doi.org/10.1371/journal.pone.0153286
10. Mannino, R.G., Myers, D.R., Tyburski, E.A. et al. Smartphone app for non-invasive detection of anemia using only patient-sourced photos. Nat Commun 9, 4924 (2018). https://doi.org/10.1038/s41467-018-07262-2
11. R. Mala, V. Anukarani, Sivani.M, G. Vaishnavi, et al. Non-Invasive Measurement of Hemoglobin Using Smartphone. Vol 44 No. 7 Journal of Harbin Engineering University ISSN: 1006-7043 [CrossRef]
12. Wemyss TA, Nixon-Hill M, Outlaw F, Karsa A, Meek J, et al. (2023) Feasibility of smartphone colorimetry of the face as an anaemia screening tool for infants and young children in Ghana. PLOS ONE 18(3): e0281736. https://doi.org/10.1371/journal.pone.0281736
13. Dimauro, G.; Simone, L. Novel Biased Normalized Cuts Approach for the Automatic Segmentation of the Conjunctiva. Electronics 2020, 9, 997. https://doi.org/10.3390/electronics9060997

14. Sabrina Dhalla, Junaid Maqbool, Tanvir Singh Mann, Aastha Gupta, Ajay Mittal, Preeti Aggarwal, Krishan Saluja, Munish Kumar, Shiv Sajan Saini, Semantic segmentation of palpebral conjunctiva using predefined deep neural architectures for anemia detection, Procedia Computer Science, Volume 218, 2023, Pages 328-337, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2023.01.015
15. https://opencv.org/
16. https://imgaug.readthedocs.io/en/latest/
17. An intelligent non-invasive system for automated diagnosis of anemia exploiting a novel dataset Artif Intell Med, 136 (2023), Article 102477, 10.1016/j.artmed.2022.102477
18. https://www.who.int/data/nutrition/nlis/info/anaemia
19. https://docs.google.com/spreadsheets/d/1AS5ZnELQjwgFQHvDR6foyp24NuUA9QlBRnYjk8jh5mA/edit#gid=0
20. https://pycaret.org/
21. https://pycaret.readthedocs.io/en/latest/api/classification.html