Multivariate Linear Regression

Nail Bed Images

Nov 14, 2022

1. Introduction

Anemia is as a life-threatening disease that affects 2 billion, or approximately 1 in 3 people world wide (Mannino 2018). In 2019, this long lasting disease accounted for 50.3 million YLD (years lived with disabilities) (Safiri 2021). Patients suffering from chronic anemia require frequent monitoring of indicators such as the Hgb concentration in blood to track the progression of their disease. Despite high prevalence, however, the current diagnostic process requires blood tests which causes discomfort and trauma in patients, in addition to incurring high monetary cost, especially in regions of lower socio-economic development (Safiri 2021). According to studies in cost-effectiveness of anemia screening, total costs per life saved in targeted screening amounted to $3575, while treating anemia can cost between $18 and $500 per month depending on the type of anemia and necessary treatment (Nosratnejad 2014). Therefore, we aim to create machine learning models that enable non-invasive inexpensive diagnosis using patients' nail bed images to predict their Hgb concentration.

Previous work focuses on predicting Hgb concentration on three regressions of interest: the fingernail beds, the conjunctiva and the palmar creases. We aim to extend the previous research by developing a multilinear LASSO regression model, using features extracted from the nail bed images as predictors and cross validation for hyperparameter tuning. We also compared our results with previous work to examine the correlation between the features and the hemoglobin concentration levels.

2. Methodology: LASSO Multilinear Model

Data

We examine 72 of the images collected from patients enrolled in Dr. Nirmish Shah's clinic. Each patient has four images corresponding to different fingers. The images are processed such that the nail bed is captured in a bounding box while the background of the image is discarded. Colour information of each pixel is extracted from the bounded nail bed images as features.

We know that each pixel can be represented by RGB values, but the RGB colour space contains both colour information and the light information, which is different for each image since photos are taken at different times and settings. To eliminate the inconsistency caused by variation in lightning and background, we used two other colour spaces (HSV, LAB) to separate the colour information from the lightning information. We computed the mean of each value/channel across the bounded nail bed image for each of these three colour spaces (HSV, LAB, RGB) and used them as our model input. Our response variable is blood hemoglobin concentration associated with each nail bed in g/dL. Our predictor variables include:

Variable Name	Description
Mean value of Hue	Average value of Hue component (the color component / base pigment) of the Hue-Saturation-Value color space
Mean value of Saturation	Average value of Saturation component (amount of color / depth of the pigment / dominance of hue) of the Hue-Saturation-Value color space
Mean value of Value	Average value of Value component (brightness of the color) of the Hue-Saturation-Value color space

Variable Name	Description
Mean value of Lightness	Average value of Lightness component from black to white on a scale of 0 - 100 of the LAB color space
Mean Value of A	Average value of representation of greenness to redness on a scale of -128 to +127 of LAB color space
Mean Value of B	Average value of representation of blueness to yellowness on a scale of -128 to +127 of LAB color space
Mean Value of R	Average value of redness of Red-Green-Blue color space
Mean Value of G	Average value of greenness of Red-Green-Blue color space
Mean Value of B	Average value of blueness of Red-Green-Blue color space

Since the dataset is private information of the patients enrolled is Dr. Shah' clinic, we would not include the details of the data in this analysis. We provide a glimpse of the data below.

```
## Rows: 72
## Columns: 15
## $ Image_URL     <chr> "https://storage.labelbox.com/cklyhytg2ulao0774uvyw0poy%~
## $ xmin          <dbl> 248, 340, 469, 644, 347, 511, 640, 744, 201, 322, 458, 6~
## $ xmax          <dbl> 292, 396, 527, 696, 385, 553, 682, 774, 248, 376, 509, 6~
## $ ymin          <dbl> 537, 465, 414, 426, 398, 413, 472, 532, 767, 796, 700, 4~
## $ ymax          <dbl> 582, 523, 474, 482, 461, 472, 522, 570, 803, 838, 726, 5~
## $ Mean_H        <dbl> 8.1922, 13.0864, 12.3233, 13.1862, 11.3486, 11.9142, 10.~
## $ Mean_S        <dbl> 0.2842, 0.2684, 0.2863, 0.2962, 0.2670, 0.2686, 0.2695, ~
## $ Mean_V        <dbl> 0.7824, 0.8003, 0.7863, 0.7895, 0.7871, 0.7820, 0.7820, ~
## $ Mean_L        <dbl> 68.7689, 68.3792, 68.0705, 68.5229, 68.4438, 68.2958, 67~
## $ Mean_A        <dbl> 15.4397, 13.9356, 15.3458, 14.7066, 14.6354, 14.0090, 14~
## $ Mean_B        <dbl> 10.6895, 12.0016, 11.8946, 14.3816, 11.9706, 11.9289, 11~
## $ Mean_Prop_R   <dbl> 0.3999, 0.3980, 0.3943, 0.3994, 0.3972, 0.3997, 0.3977, ~
## $ Mean_Prop_G   <dbl> 0.3077, 0.3125, 0.3147, 0.3140, 0.3113, 0.3115, 0.3107, ~
## $ Mean_Prop_B   <dbl> 0.2925, 0.2895, 0.2910, 0.2866, 0.2915, 0.2888, 0.2916, ~
## $ concentration <dbl> 9.5, 9.5, 9.5, 9.5, 9.5, 9.5, 9.5, 9.5, 9.2, 9.2, 9.2, 9~
```

Exploratory Data Analysis

concentration
Min. : 9.20
1st Qu.: 9.50
Median : 9.90
Mean :10.43
3rd Qu.:10.20
Max. :13.30

Figure 1 and the corresponding summary statistics show that the Hemoglobin concentration in g/dL ranges from 9.20 g/dL to 13.30 g/dL, with a mean of 10.43 g/dL and median of 9.90 g/dL, which better captures the center of the distribution since it is right-skewed.

Cross Validation for Hyperparameter Tuning

In order to perform variable selection to gauge insight into the predictors which have significantly associations with hemoglobin concentrations, we use the LASSO which uses L_1 norm penalty, shrinking the coefficient estimates of insignificant predictors towards zero.

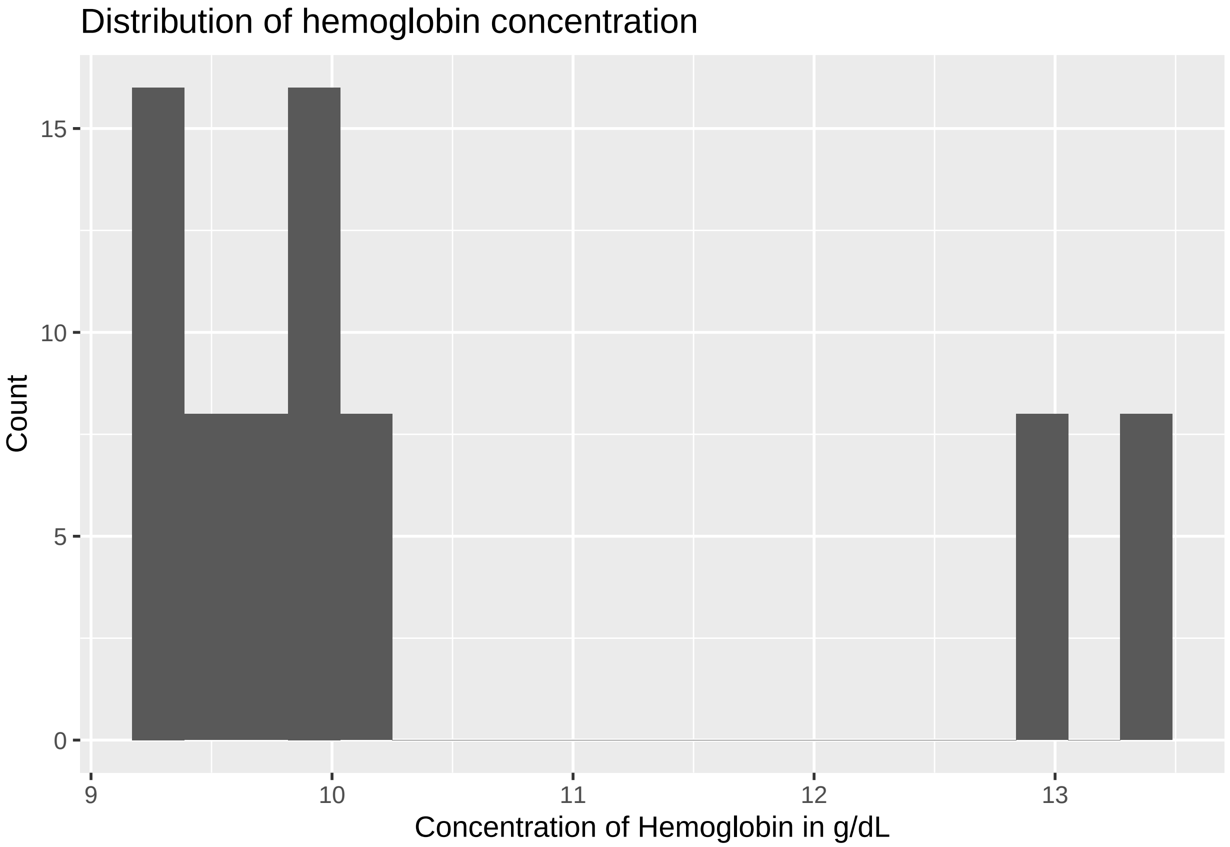
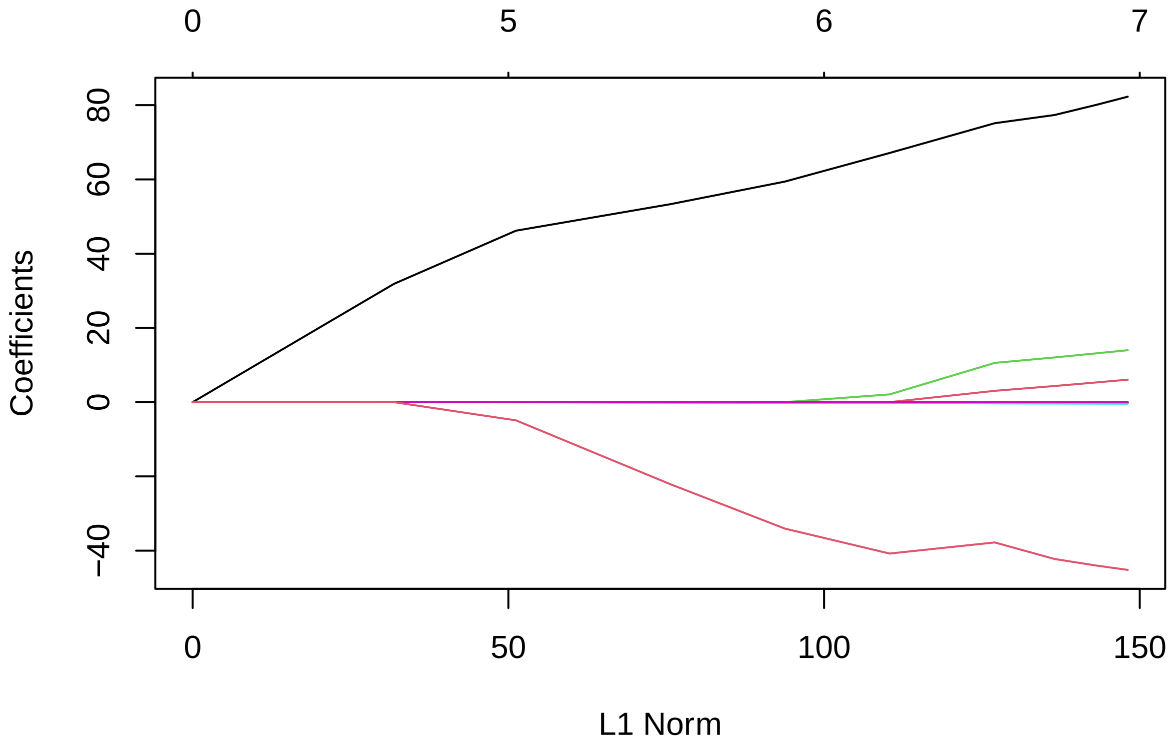


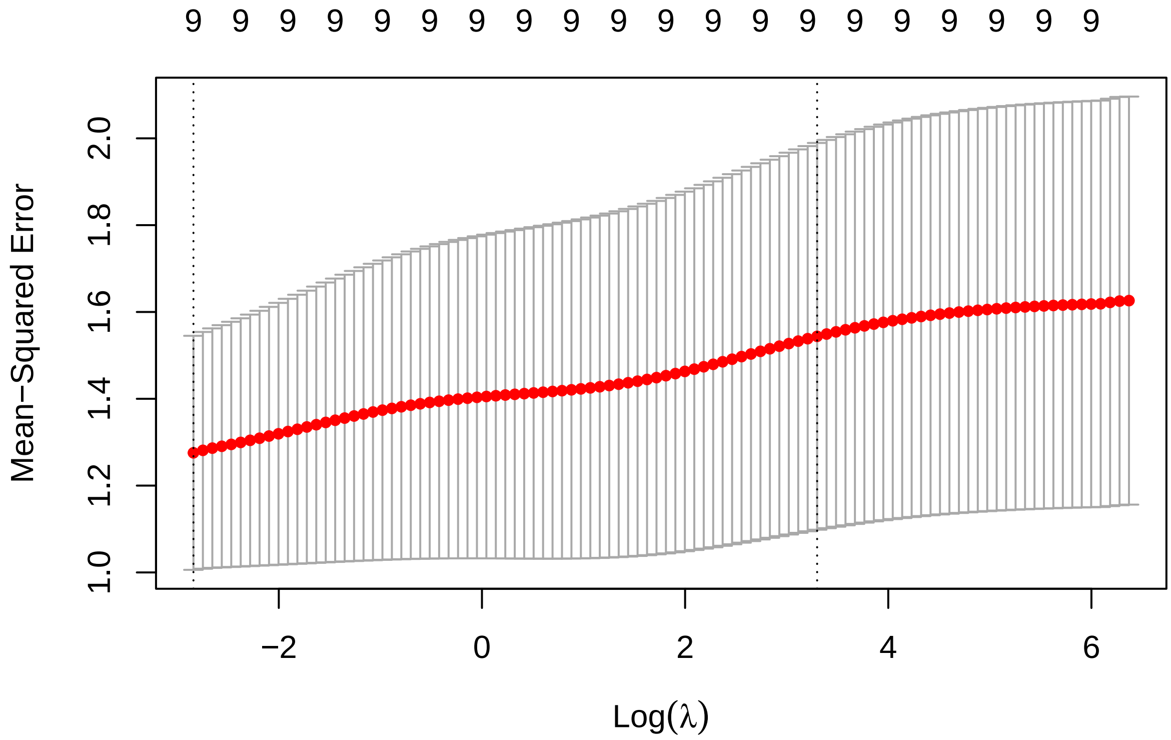
Figure 1: Distribution of response variable

We use cross validation to tune the hyperparamter λ and visualize the shrinkage of the coefficients:

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



We observe that the best λ which results in the smallest MSE is 0.0584



```
## [1] 0.05840522
```

	s0
(Intercept)	15.4670
Mean_H	0.0045
Mean_S	11.6495
Mean_L	0.0124
Mean_Prop_G	-30.2895

As a result, we derive a sparse model which only involves a subset of the features extracted from the nailbed images.

Predictions

```
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test,])
mean((lasso.pred - y.test)^2)
```

```
## [1] 14.01923
```

Performing predictions on the test data set, we derive a test MSE of 14.019.

Discussion of Model Output

We have derived the following linear model:

$$\hat{H}gb = 15.4670 + 0.0045 \times Mean\_H + 11.6495 \times Mean\_S + 0.0124 \times Mean\_L - 30.2895 \times Mean\_Prop\_G$$

We notice that while variables `Mean_L`, `Mean_A`, `Mean_B`, `Mean_Prop_R`, `Mean_Prop_G`, and `Mean_Prop_B` are negatively associated with the response variable `Hgb` concentration, `Mean_H`, `Mean_S` and `Mean_V` are positively associated with the response variable. This means as mean value of hue, or mean value of saturation, or mean value of brightness of HSV color space increases, the hemoglobin concentration level also tends to increase. On the other hand, as the mean value of brightness, redness, greenness or blueness of RGB color space increases, the hemoglobin concentration level tends to decrease.

We find that out of all the predictors, `Mean_S` tends to associates most significantly with the response variable with the smallest p value. For each 1 unit increase in the mean value of saturation, Hgb concentration is expected to increase by 31.719 on average, keeping all else constant. The second most significant predictor is `Mean_A`, by which we found that for each 1 unit increase in the mean value of representation of greenness to redness, Hgb concentration is expected to decrease by 0.435 on average, keeping all else constant.

2.2 Principle Components Analysis (PCA)

Data

In the first part of our research, we used the TBND_V2 (Transient Biometrics Nails Dataset) on Kaggle

1

, which contains unlabeled nail bed images (Barbosa, Theoharis, and Abdallah 2019).

Experiment on TBND_V2 Dataset

We initially experimented with the TBND_V2 dataset found on Kaggle, since there is not enough data from the patients. Given that the data is unlabelled, we tried unsupervised clustering methods on the data, hoping to gain some insights on classification of nail bed images.

To get features, we used VGG16, a convolutional neural network (CNN) in our model. It extracts features from the input images, turning each image into feature vectors (4096 by 1). We removed the final (prediction) layer from the neural network manually, and the new output layer is a fully-connected layer with 4,096 individual nodes. We do this by specifying the “outputs” argument when initialising the model. We therefore get input of our model by using the neural net VGG16 as a feature extractor for the image data.

We then performed a principal component analysis (PCA) on the feature vectors to reduce the dimension of the feature space. For each of the 93 image samples, we now have a corresponding 1 by 4096 feature vector. This means that our model needs to process a 93 by 4096 matrix. To reduce the computational and complexity cost of processing high-dimensional data, we performed principal component analysis (PCA) on the matrix for dimension reduction. We set the parameter to 50 to obtain the top 50 principal components of the feature vector. The principal components are by default sorted in descending order. This means that the first principal component will be able to explain the most variability in the feature vector. It’s a linear combination of the feature variables, and its direction captures the most variability. Thus, PCA helps us to reduce the dimension of the features from 4096 to 50 while preserving as much information in the original data as possible.

After getting features, we used Kmean clustering, an unsupervised algorithm that is commonly used in exploratory data analysis, to perform the clustering. The Kmean clustering works as follows: Initialise the centre of each k cluster by shuffling the dataset and randomly selecting K data points without replacement. Iterate the following steps until the assignment of data points to clusters is no longer changing: Compute the sum of the Euclidean distance squared between data points and all centres. Assign each data point to the closest cluster; each cluster is represented by its unique centre point. Compute the cluster’s centroid by taking the average of all data points assigned to that cluster.

In our model, we set the hyperparameter k to be 5 and clustered all samples into 5 categories. Based on the features we extracted, each cluster will contain images that are visually similar.

PCA Results

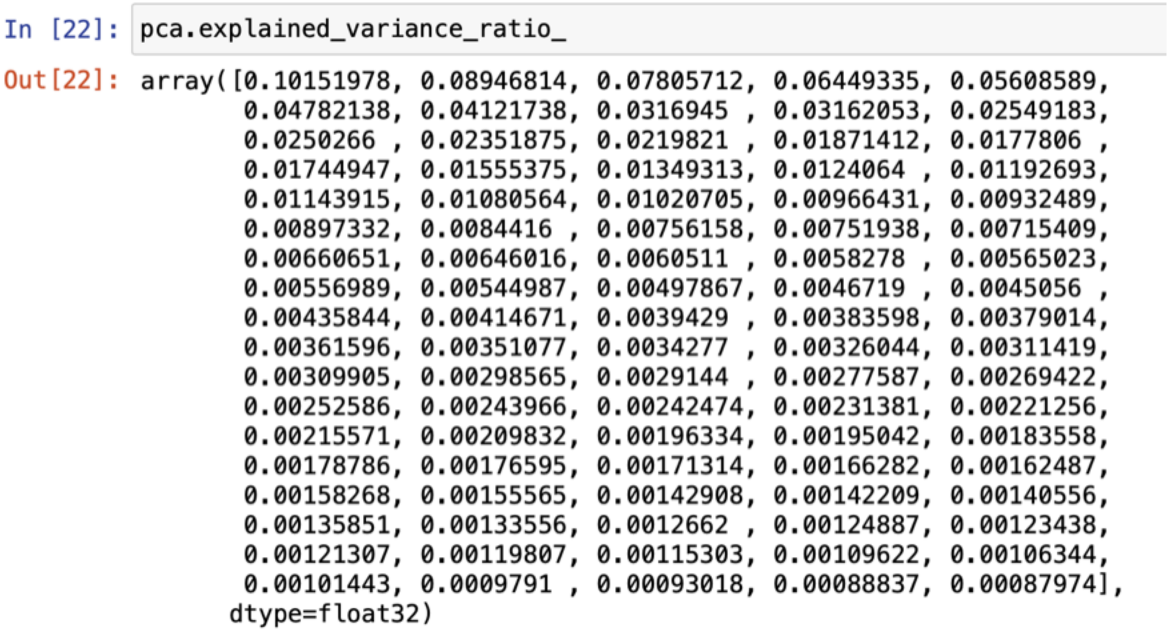
Based on the documentation, the explained_variance_ratio_ function returns the percentage of variability explained by each of the selected components. Running this function gives us the amount of variability that is explained by all the PCs (0.1015 is explained by the first PC, 0.0894 by the second, and 0.0781 by the third etc.)

Then we generated a bar chart to represent the variability explained by different principal components, as well as the cumulative step plot to represent the variability explained by the first most important components.

The neural net in the model functions as a feature extractor for the image data, which is the input of our model. To be more specific, we used VGG16, a convolutional neural network (CNN) in our model. It extracts features from the input images, turning each image into feature vectors (4096 by 1). We removed the final (prediction) layer from the neural network manually, and the new output layer is a fully-connected layer with 4,096 individual nodes. We do this by specifying the “outputs” argument when initialising the model.

The PCA reduces dimensions of our feature vectors. For each of the 93 image samples, we now have a corresponding 1 by 4096 feature vector. This means that our model needs to process a 93 by 4096 matrix. To reduce the computational and complexity cost of processing high-dimensional data, we performed principal component analysis (PCA) on the matrix for dimension reduction. We set the parameter to 50 to obtain the top 50 principal components of the feature vector. The principal components are by default sorted in descending order. This means that the first principal component will be able to explain the most variability in the feature vector. It’s a linear combination of the feature variables, and its direction captures the most variability. Thus, PCA helps us to reduce the dimension of the features from 4096 to 50 while preserving as much information in the original data as possible.

In our model, we set the hyperparameter k to be 5 and clustered all samples into 5 categories. Based on the features we extracted, each cluster will contain images that are visually similar.



```
In [22]: pca.explained_variance_ratio_

Out[22]: array([0.10151978, 0.08946814, 0.07805712, 0.06449335, 0.05608589,
                0.04782138, 0.04121738, 0.0316945 , 0.03162053, 0.02549183,
                0.0250266 , 0.02351875, 0.0219821 , 0.01871412, 0.0177806 ,
                0.01744947, 0.01555375, 0.01349313, 0.0124064 , 0.01192693,
                0.01143915, 0.01080564, 0.01020705, 0.00966431, 0.00932489,
                0.00897332, 0.0084416 , 0.00756158, 0.00751938, 0.00715409,
                0.00660651, 0.00646016, 0.0060511 , 0.0058278 , 0.00565023,
                0.00556989, 0.00544987, 0.00497867, 0.0046719 , 0.0045056 ,
                0.00435844, 0.00414671, 0.0039429 , 0.00383598, 0.00379014,
                0.00361596, 0.00351077, 0.0034277 , 0.00326044, 0.00311419,
                0.00309905, 0.00298565, 0.0029144 , 0.00277587, 0.00269422,
                0.00252586, 0.00243966, 0.00242474, 0.00231381, 0.00221256,
                0.00215571, 0.00209832, 0.00196334, 0.00195042, 0.00183558,
                0.00178786, 0.00176595, 0.00171314, 0.00166282, 0.00162487,
                0.00158268, 0.00155565, 0.00142908, 0.00142209, 0.00140556,
                0.00135851, 0.00133556, 0.0012662 , 0.00124887, 0.00123438,
                0.00121307, 0.00119807, 0.00115303, 0.00109622, 0.00106344,
                0.00101443, 0.0009791 , 0.00093018, 0.00088837, 0.00087974],
               dtype=float32)
```

Figure 2: PCA variance ratios

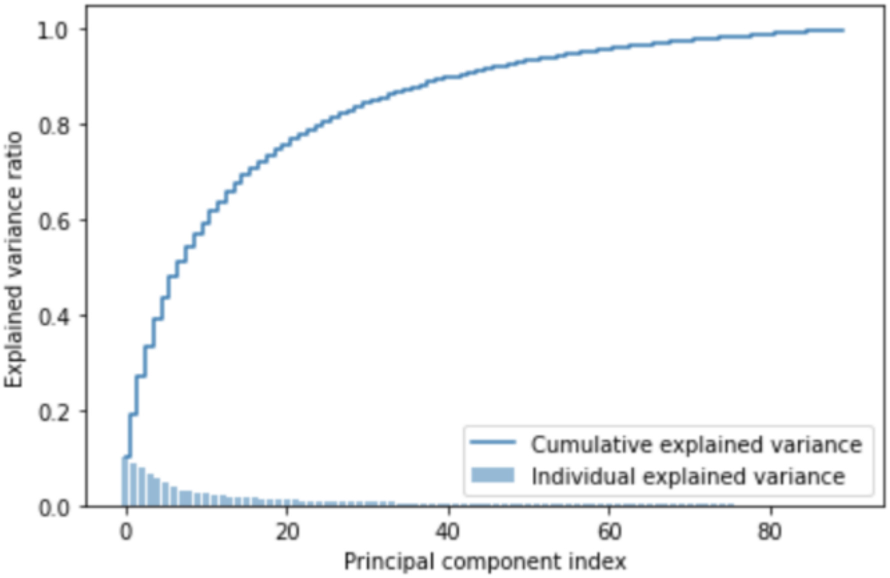


Figure 3: PCA cumulative step plot

3. Conclusion

In this study, we explored primarily two methods in predicting hemoglobin concentration level based on parameters derived from patients' nail beds images: (1) multilinear regression, (2) principle components analysis. While the first model provides reasonable train and and test error, the p value of the predictors seems to be large. Our second model yields a reasonable explained variance ratio with respect to the scale of principle components. Limitations of our work includes that the data from the clinic lacks the label, therefore we are unable to use classification methods such as random forest or SVM. Our future work includes fitting Bayesian regressions on the data, exploring more machine learning methods such as random forest or support vector machines, and using shrinkage methods such as ridge regression and LASSO.

Appendix

References

Barbosa, Igor Barros, Theoharis Theoharis, and Ali E. Abdallah. 2019. "TBND_V2." Kaggle. https://doi.org/10.34740/KAGGLE/DS/309682.

Mannino, Robert G. 2018. "A NONINVASIVE, IMAGE-BASED SMARTPHONE APP FOR DIAGNOSING ANEMIA," 12.

Nosratnejad, Barfar, S. 2014. "Cost-Effectiveness of Anemia Screening in Vulnerable Groups: A Systematic Review." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4124557/.

Safiri, Saeid. 2021. "Burden of Anemia and Its Underlying Causes in 204 Countries and Territories, 1990–2019: Results from the Global Burden of Disease Study 2019," 1. https://jhoonline.biomedcentral.com/articles/10.1186/s13045-021-01202-2.