MAY 17, 2020

PROJECT REPORT

SKIN SEGMENTATION DATASET

EE257 – MACHINE LEARNING

ROJIN ZANDI
014491256

1.0 Dataset Description:

Skin Segment dataset has been collected from various people with different races, ages and genders. The data provides B (Blue), G (Green), R (Red) color space as *features* and Class label as *output* which are the columns of the file. There are 245057 data points at dataset that 50859 of them are *skin* samples (Class=1) and 194198 are *non-skin* samples (Class=2).
There are no outliers and missing data in the dataset.

	Blue	Green	Red	C
Count	245057	245057	245057	245057
Mean	125.065446	132.507327	123.177151	1.792461
Std	62.255653	59.941197	72.562165	0.405546
Variance	3875.7663	3592.94119	5265.26774	0.164468
Min	0.0	0.0	0.0	1.0
25%	68.0	87.0	70.0	2.0
50%	139.0	153.0	128.0	2.0
75%	176.0	177.0	164.0	2.0
max	255.0	255.0	255.0	2.0

Table 1.1: Dataset Description

2.0 Data Visualization

The collected data can be shown in different diagram. The histogram, scatter and box plot of the dataset are provided below.

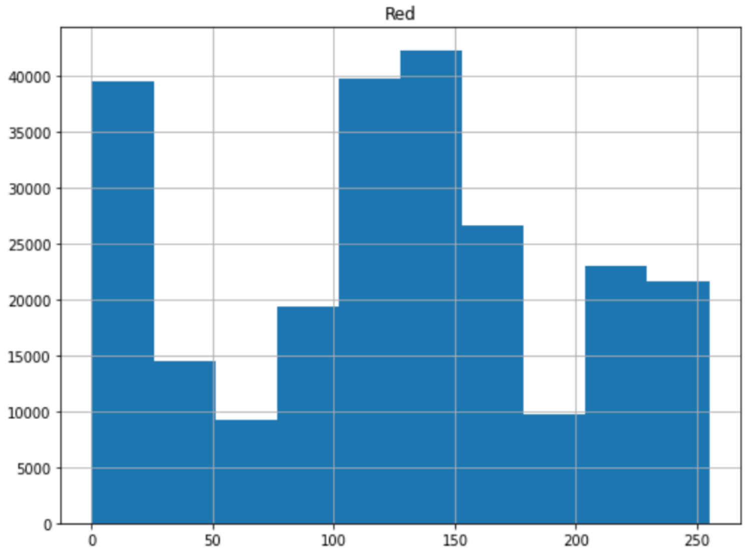

Fig. 2.1: The histogram plot of Red feature. It is bimodal.

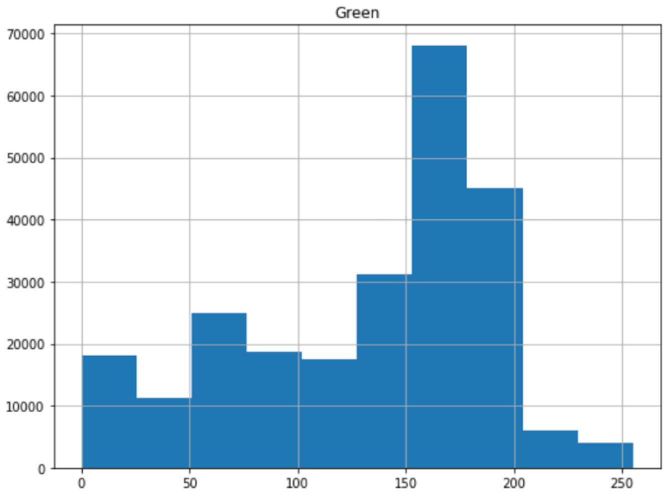

Fig. 2.2: The histogram plot of Green feature. It is unimodal and right skewed.

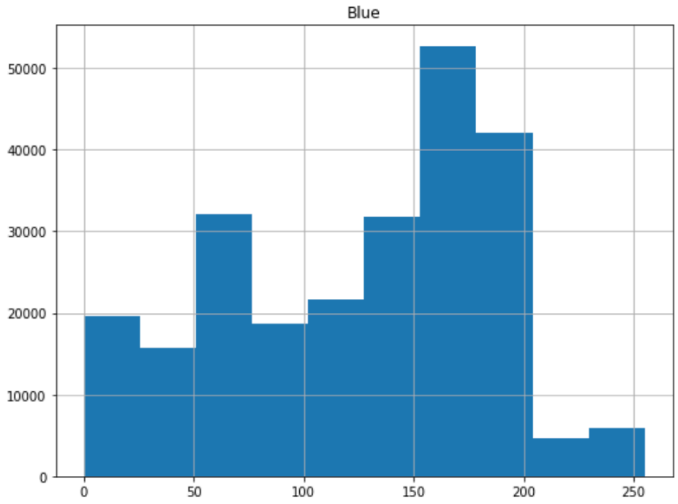

Fig. 2.3: The histogram plot of Blue feature. It is unimodal and right skewed.

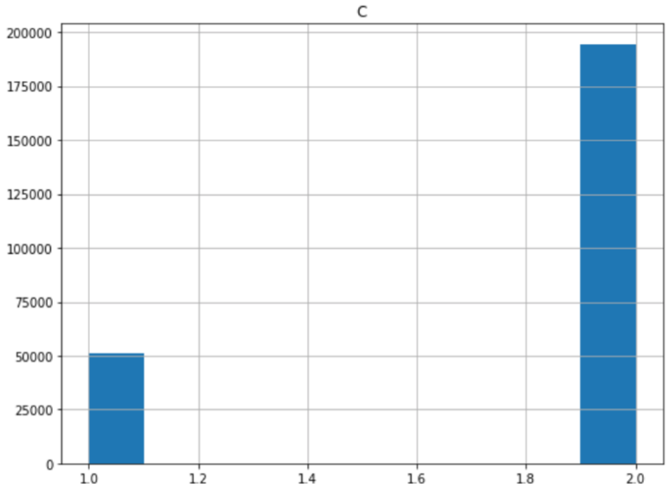

Fig. 2.4: The histogram plot of output. The data is unbalanced.

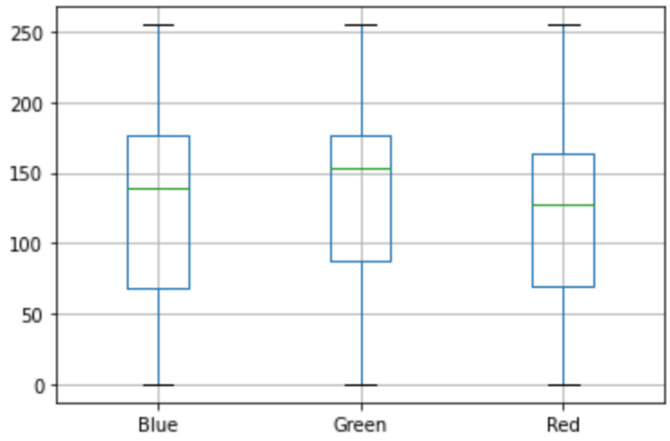

Fig. 2.5: The box plot of features. The green line shows the 50% of the data.

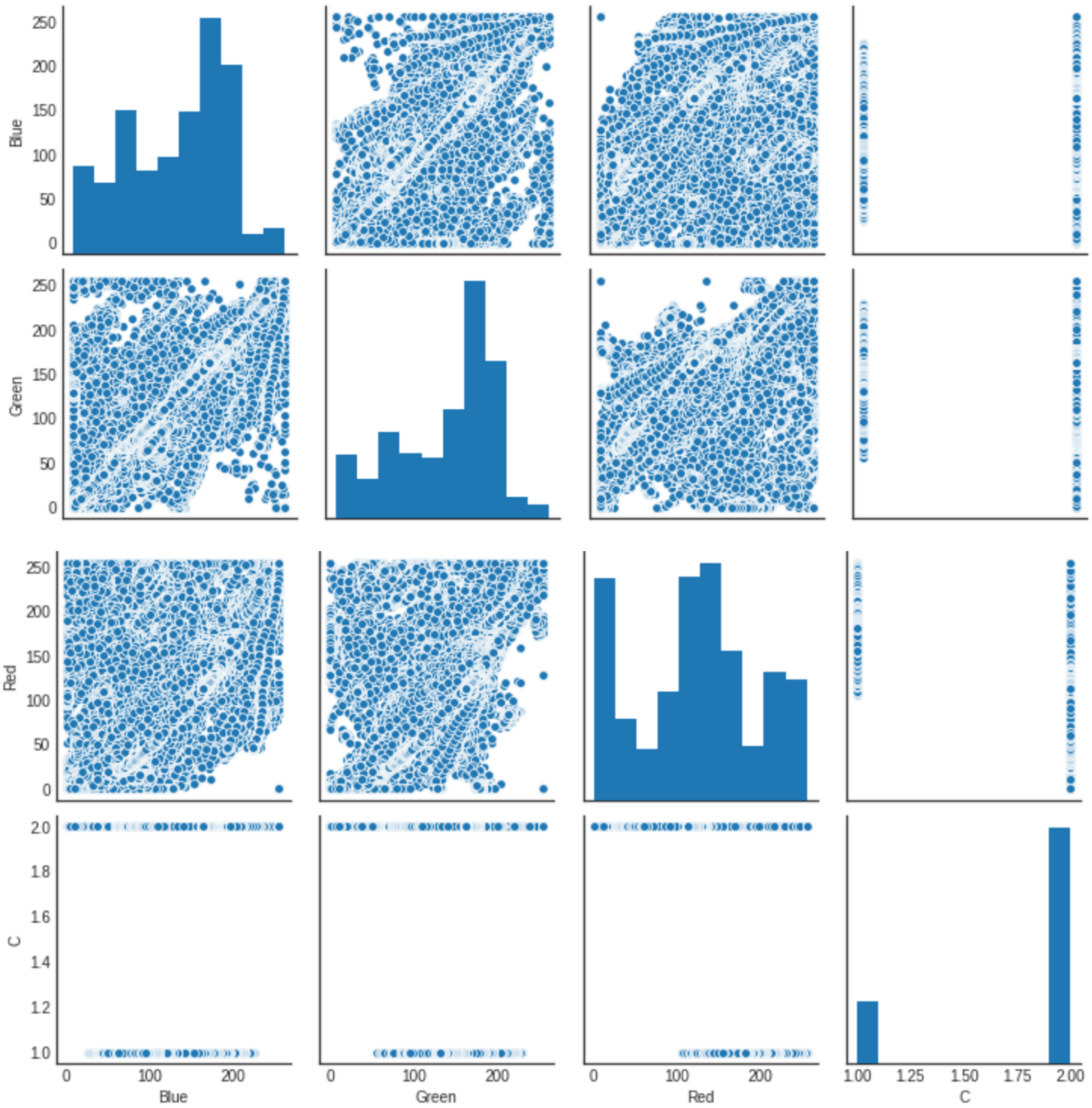

Fig 2.6: Scatter plot of features (Blue, Red, Green) and output (C)

Principle Component Analysis (PCA) is an unsupervised learning method which finds low dimensional representation of the data. This method is also used for visualization and it illustrates which features are more correlated. In order to find the first principle component, we need to find direction with largest variance. The second principle component will be in a direction with second largest variance and also it must be orthogonal to the first principle component. The provided code contains four principle components of the skin segmentation dataset. (Table 2)

	V1	V2	V3	V4
Blue	-0.532786	-0.438430	-0.320739	0.648880
Green	-0.594468	-0.235527	-0.195104	-0.743687
Red	-0.553872	0.332306	0.750345	0.140646
C	0.236587	-0.801174	0.544098	-0.078126

Table 2.1: The principle component loading vectors. These are also displayed in Figure 2.7.

Figure 2.7 plots the first two principle components of the skin segmentation dataset. As we see, our features (Blue, Green, Red) are more correlated and the first principle component puts more weight on these three, comparing to the output (C). but the second principle component is along the output vector.

Each principle component explains some amount of variance and standard deviation and we are interested in the highest variance. The table 2.2 provides the amount of explained standard deviation, variance, and variance ratio. As it was expected, the first principle component has the largest variance and standard deviation and the fourth one has the lowest variance. Figure 2.8 illustrates the proportion of variance for individual and all principle components.

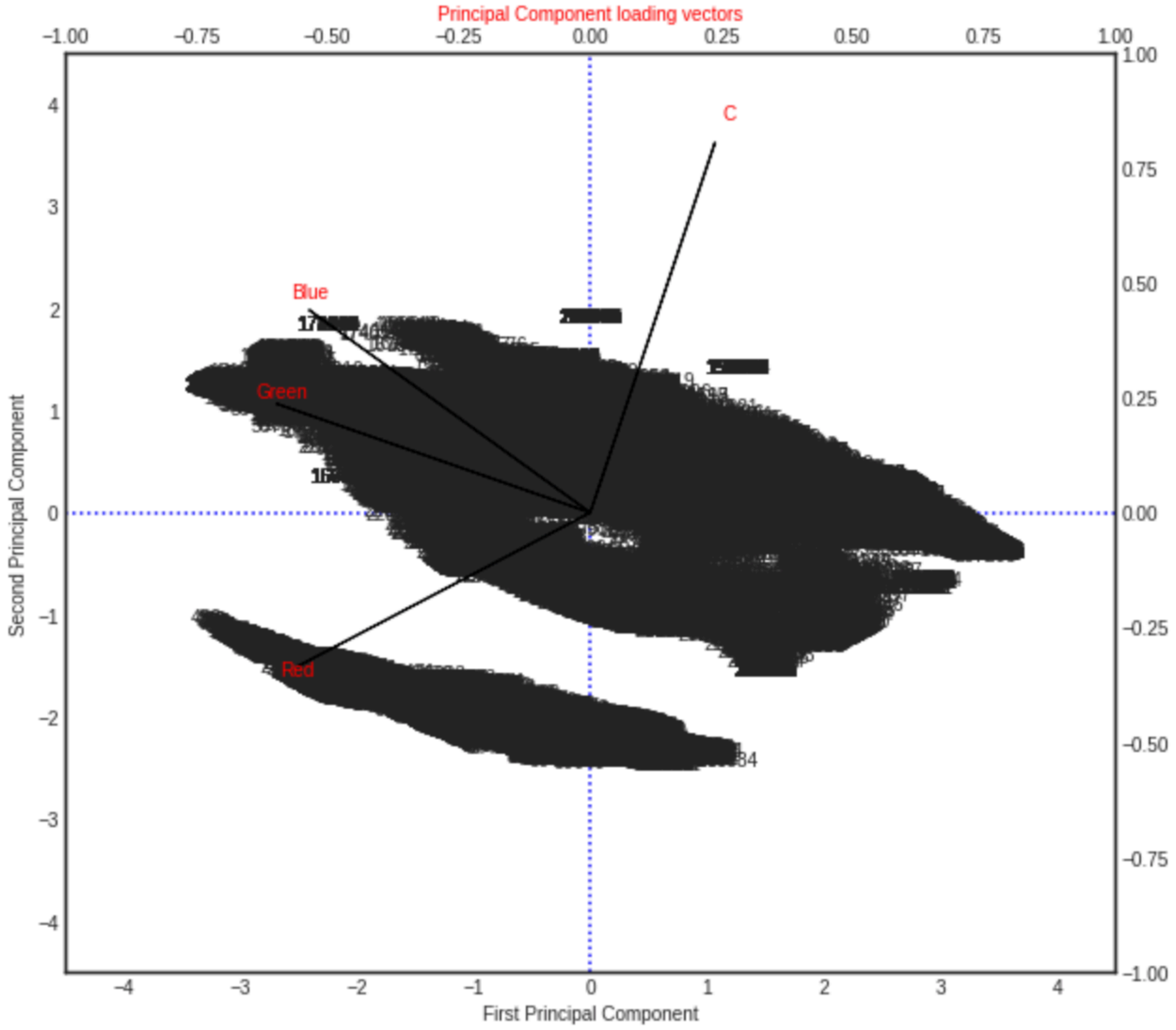

Fig 2.7: Two first Principle Components of the skin segmentation dataset

	PC1	PC2	PC3	PC4
STD Deviation	1.55866233	1.11865863	0.4543324	0.34102905
Variance	2.42943827	1.25139713	0.20289011	0.11630081
Variance Ratio	0.60735459	0.31284801	0.05072232	0.02907508

Table 2.2: The Standard Deviation, Variance and Variance Ratio of principle components.

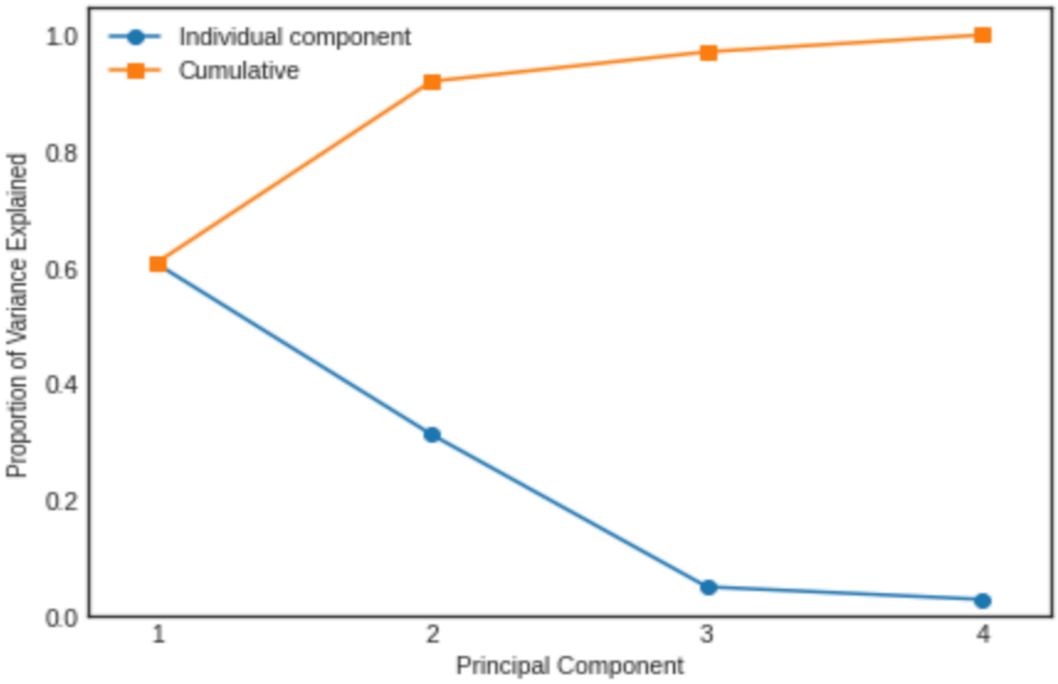

Fig 2.8: The blue line shows the proportion of variance by each component and the orange line shows the proportion of variance cumulatively. So, all the components together explain the most variance.

3.0 Dataset Cleaning

Outliers:

In order to find the outliers of the dataset, we should compute IQR (In Quartile Range) and Range. If there are any data points in the dataset, which are more than computed range, they are counted as outliers. Fortunately, there is no outliers in the dataset.

$IQR = Q_3 - Q_1$
$Range = [Q_1 - K*IQR \qquad Q_3 + K*IQR]$, where $K=1.5$

	Blue	Green	Red	C
Q_1	68	87	70	2
Q_3	176	176	164	2
IQR	108	90	94	0
Range	[-94 417]	[-48 312]	[-71 305]	[2 2]
Outliers	no	no	no	---

Table 3: The quartiles and outliers of the dataset

4.0 Related Work

4.1 Adaptive Digital Makeup

This paper aims to represent a system for digital make up. They have used skin segmentation dataset because of diversity in skin tone (based on ethnicity) and genders. There are four machine learning techniques applied in this papers that each one of them tries to classify different features. For example, Fuzzy learning decision tree to recognize skin and non-skin data. HAAR, ASM and SVM are other applied techniques in this paper. The researchers have improved the appearance of the skin region and made the process faster than the past.

4.1 Efficient Skin Region Segmentation using Low Complexity Fuzzy Decision Tree Model

The aim of this paper is to apply low complexity fuzzy decision tree on the skin segmentation dataset to make the segmentation efficient for application into embedded devices. Also, this system is fast enough to have real time performance which means it can be applied into products.

5.0 Feature Extraction

5.1 Comparing Pearson Correlation

Feature extraction helps us to choose which features are more useful to create an efficient model. If the model includes all the features, it becomes more complicated. Easiest way to choose features is to compare correlation of features and output. Higher correlation means that the feature is more effective.
Fig. 5.1 illustrates the correlation (Pearson) between features and output. Darker areas mean higher correlation.

	Blue	Green	Red	Output
Blue	1.000	0.855250	0.496376	0.092030
Green	0.855250	1.000	0.660098	-0.120327
Red	0.496376	0.660098	1.000	-0.569958
Output	0.092030	-0.120327	-0.569958	1.000

Table 5: Correlation table of features and output

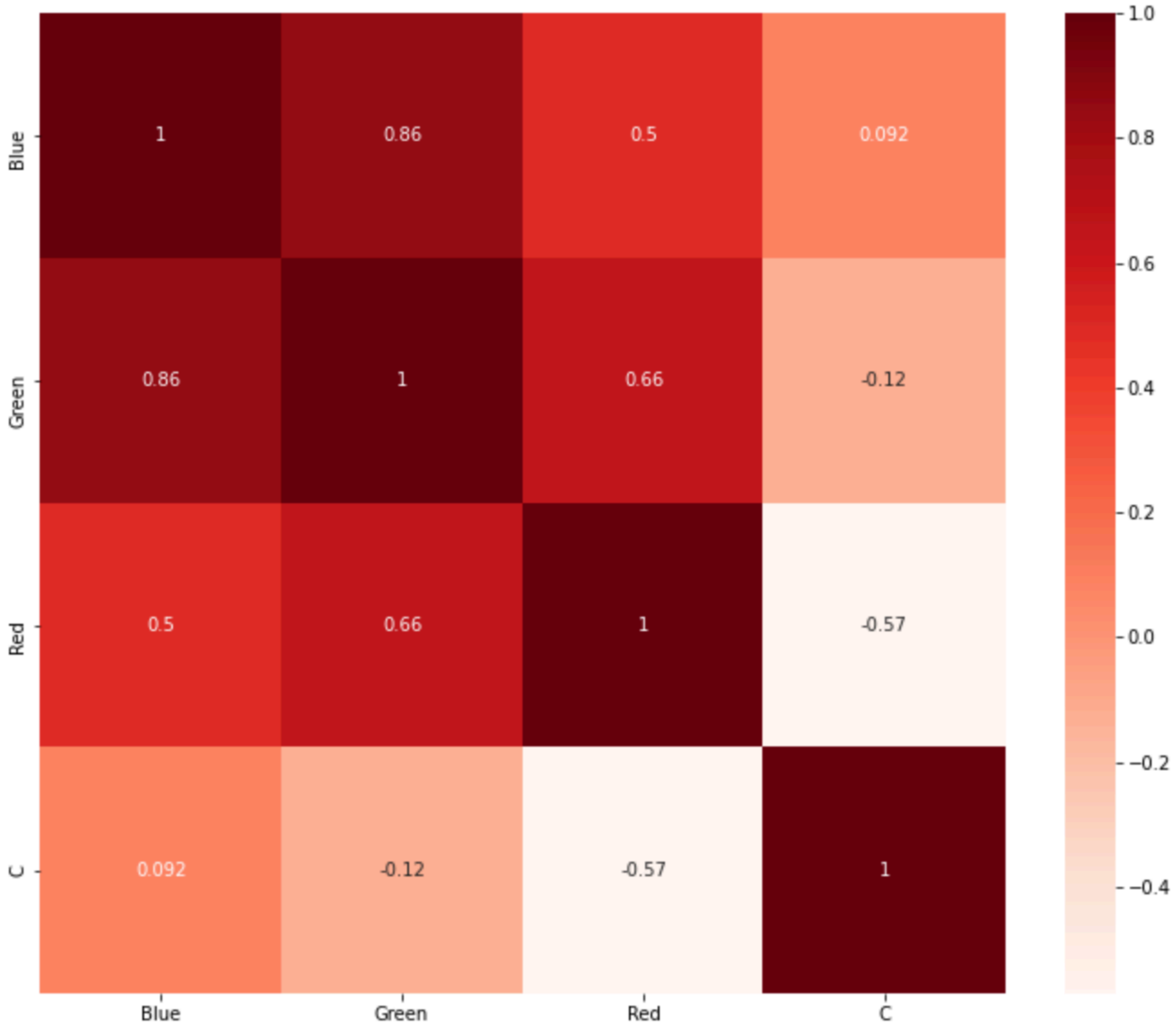

Fig. 5.1: Heatmap of correlation between feature and output

5.2 Recursive Feature Elimination

Another approach to eliminate features is *Recursive Feature Elimination*. This method computes model accuracy and score with different features and recursively removes weakest features. For the skin segmentation dataset, the recursive feature elimination suggests keeping all three features (Blue, Green, Red). The provided code shows True for all the features.

5.3 Lasso regression

Lasso regression also can eliminate features by decreasing their coefficients to zero. The purpose of this method is decreasing MSE of the model, so it puts a penalty term to find the best coefficients (β_j) for the model. Lasso regression is an optimization problem which is in the form of:

$$Minimize \ \{\textstyle\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \ + \ \lambda \sum_{j=1}^{p} |\beta_j| \qquad (1)$$

Where p is the number of our features and λ is the tuning parameter which controls the trade-off between error and largeness of the coefficients. Figure 5.2 illustrates the effect of increasing λ on the test and train error. Based on this figure we select best tuning parameter and then tune the coefficients. As provided in the code, the best coefficients -chosen by Lasso Regression- are:

Blue	*0.208702*
Green	*-0.011182*
Red	*-0.326028*

And MSE of the model with these coefficients is 0.07957137853234084.

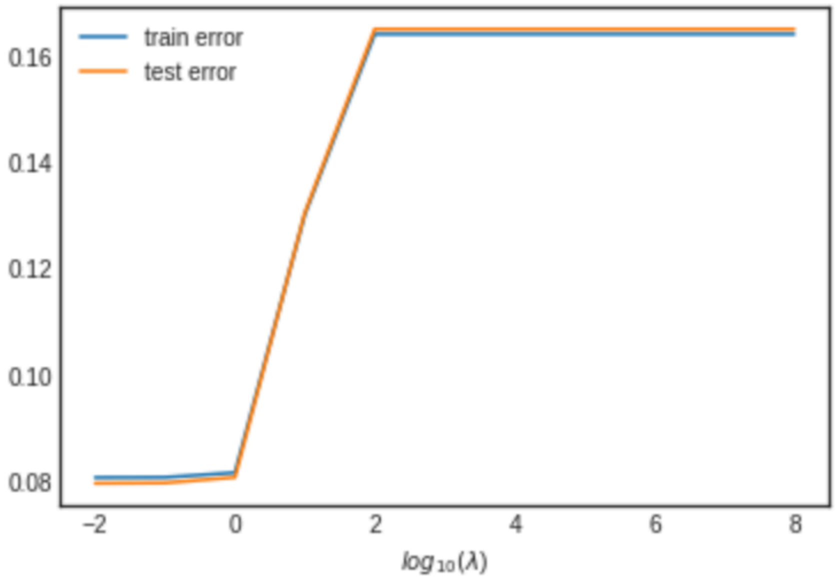

Fig. 5.2: The Y axis shows the error and the X axis shows the largeness of tuning parameter.

Notes: Ridge regression is a fine-tuning process but because of closeness to Lasso, the code has been done in this part.

6.0 Model Development:

There are various classification models than can be applied to the skin segmentation dataset. We are going to apply five different models.

6.1 Logistic Regression

The performance and metrics are discussed in part 8.

6.2 Linear/Quadratic Discriminant Analysis

The performance and metrics are discussed in part 8.

6.3 K-Nearest Neighbor (K=1 and K=5)

The performance and metrics are discussed in part 8.

6.4 Decision Tree

To obtain the Decision Tree, Gini Index function is applied to measure the quality of the split. Each split in figure 6.1 shows the Gini measure. As the Gini Index decreases, the performance of the model increases. The orange leaves are not-skin and the blue ones show the skin region. Figure 6.2 illustrates the variable importance which uses mean decrease in Gini Index to compute the importance of each feature. As it can be seen, the Red feature is the most and the Blue is the least important feature the Decision tree.

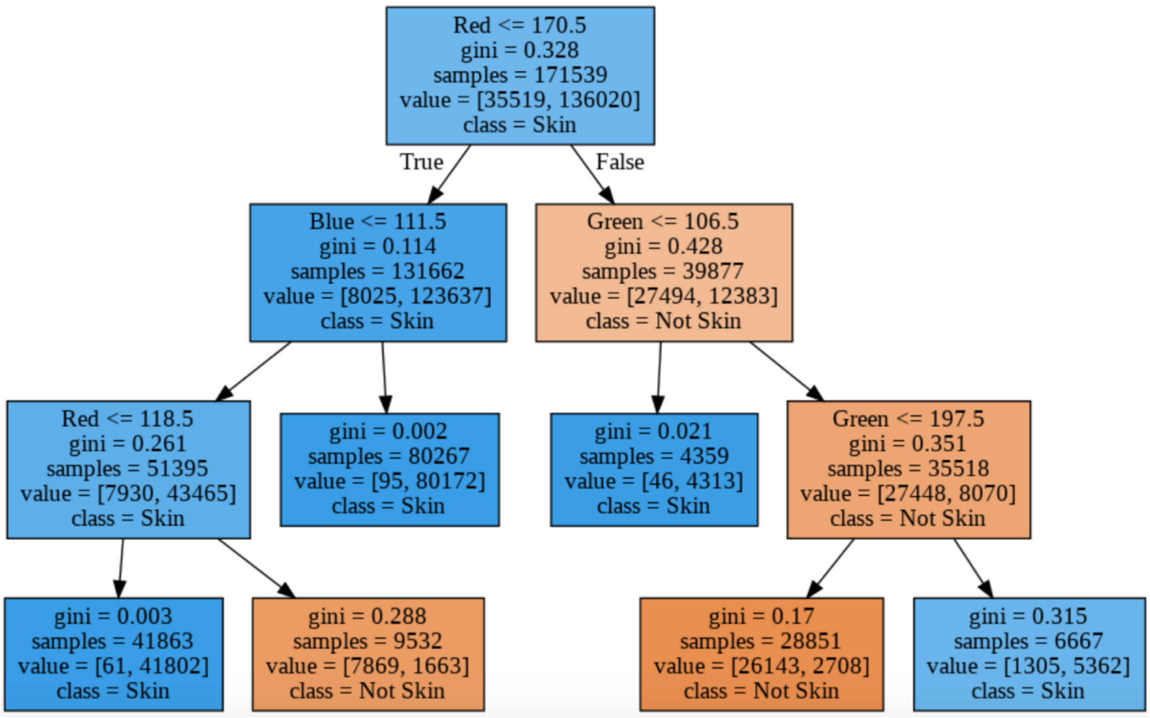

Fig 6.1: A plot of the Decision Tree of the Skin Segmentation dataset.

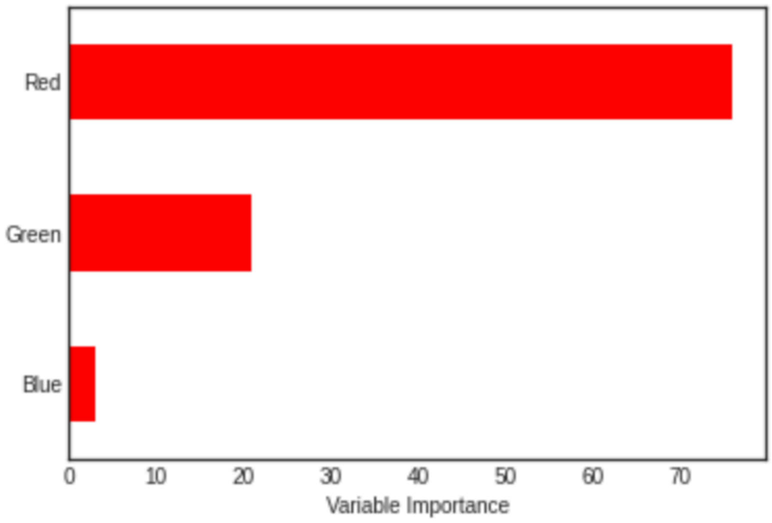

Fig 6.2: A variable importance plot for the Skin Segmentation dataset.

6.5 Random Forest

The performance and metrics are discussed in part 8.

7.0 Fine-tune your models & Feature Set:

Ridge Regression is an approach to find the best coefficients to decrease the error of the model. This approach is close to the Lasso Regression, but Ridge regression cannot eliminate the features. It can make the coefficients small by largening the tuning parameter. By applying Ridge regression on skin segmentation dataset, with two different alphas, the Table 7.1 is obtained. In the second row and Lasso Regression, alpha has been obtained by Cross Validation and the decrease in MSE is explicit. Figure 7.1 illustrates the relation between the tuning parameter and model error.

	Blue	**Green**	**Red**	**MSE**
Alpha=4	0.000005	-0.000006	-0.000028	0.1651074
Alpha=21.64	0.210125	-0.012691	-0.325927	0.0795665
Lasso Reg.	0.208702	-0.011182	-0.326028	0.0795713

Table 7.1: Obtained Coefficients and MSE by Ridge and Lasso Regression

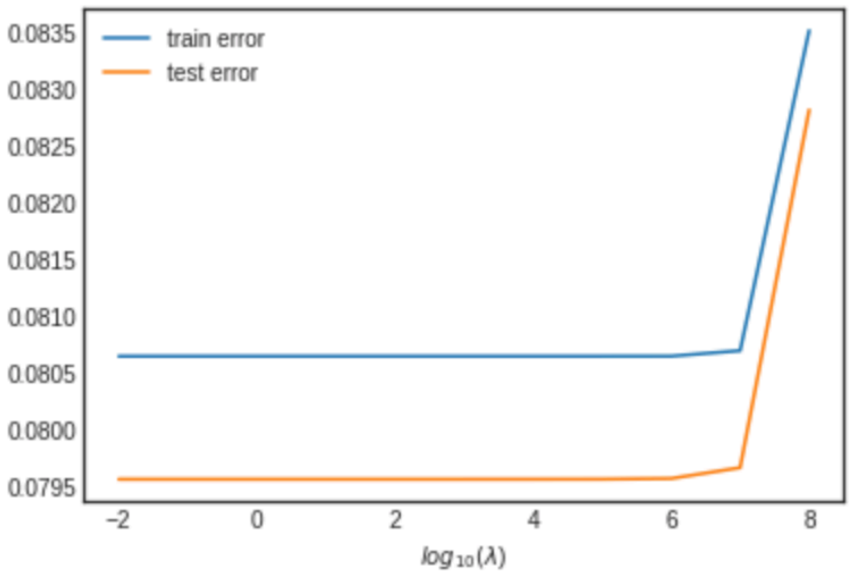

Fig. 7.1: The Y axis shows the error and the X axis shows the largeness of tuning parameter.

To tune the hyperparameters of our models, K-fold cross validation with K=5 has been done on the models. The tuned parameters for each model are:

Logistic Regression: Number of iterations

LDA/QDA: Tol

KNN: Number of Neighbors (K)

Decision Tree and Random Forest: Maximum depth of the tree

8.0 Performance:

	Precision	Recall	Accuracy	Score
Logistic Reg.	0.95	0.94	0.92	0.92029162926
LDA	0.97	0.94	0.93	0.93103026133
QDA	0.99	1.00	0.99	0.98358973761
KNN	1.00	1.00	1.00	0.99955112248
Decision Tree	0.99	0.97	0.97	0.96573373984
Random Forest	0.91	0.97	0.90	0.90101959932
Best model	KNN	KNN	KNN	KNN

Table 8.1: Comparing classification metrics for Training data before fine-tuning

	Precision	Recall	Accuracy	Score
Logistic Reg.	0.95	0.93	0.91	0.90528850077
LDA	0.97	0.94	0.93	0.93265594820
QDA	0.99	0.93	0.98	0.98399031529
KNN	1.00	0.999	0.999	0.99949672189
Decision Tree	0.99	0.97	0.97	0.965994722380

Random Forest	0.91	0.97	0.90	0.902894529230
Best model	KNN	KNN	KNN	KNN

Table 8.2: Comparing classification metrics for Test data before fine-tuning

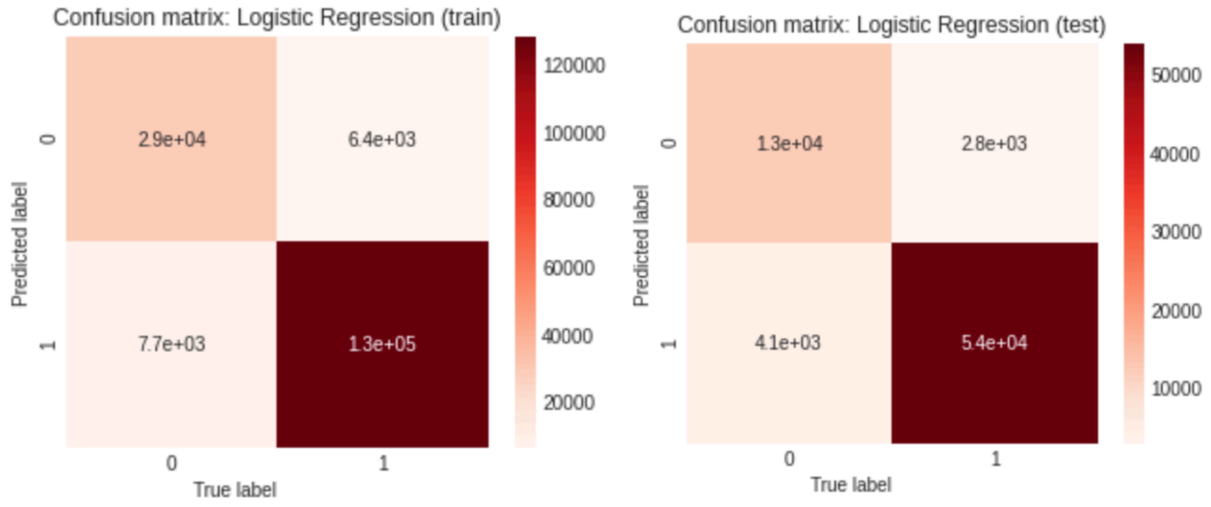

Fig. 8.1: Confusion matrix of Logistic Regression

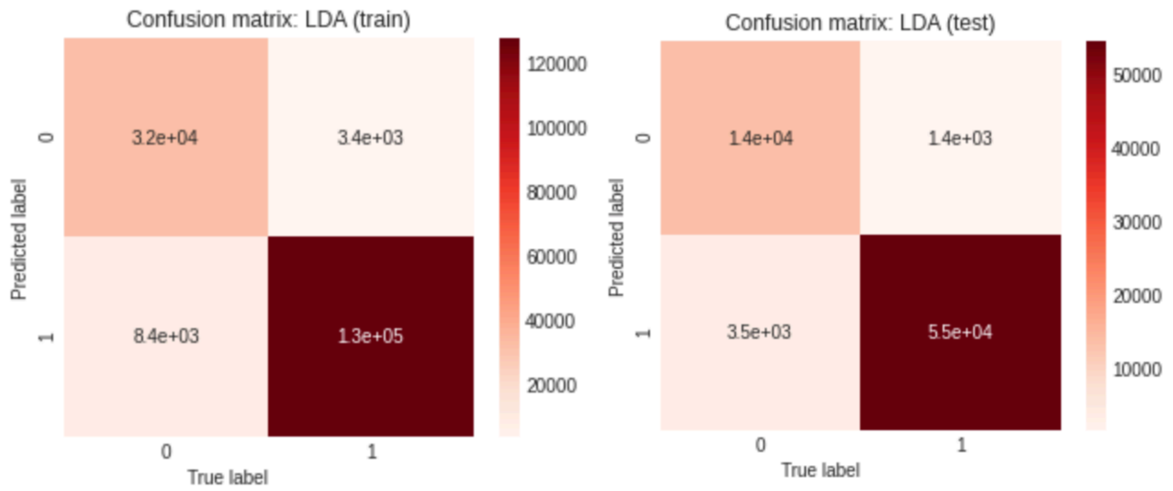

Fig. 8.2: Confusion matrix of Linear Discriminant Analysis

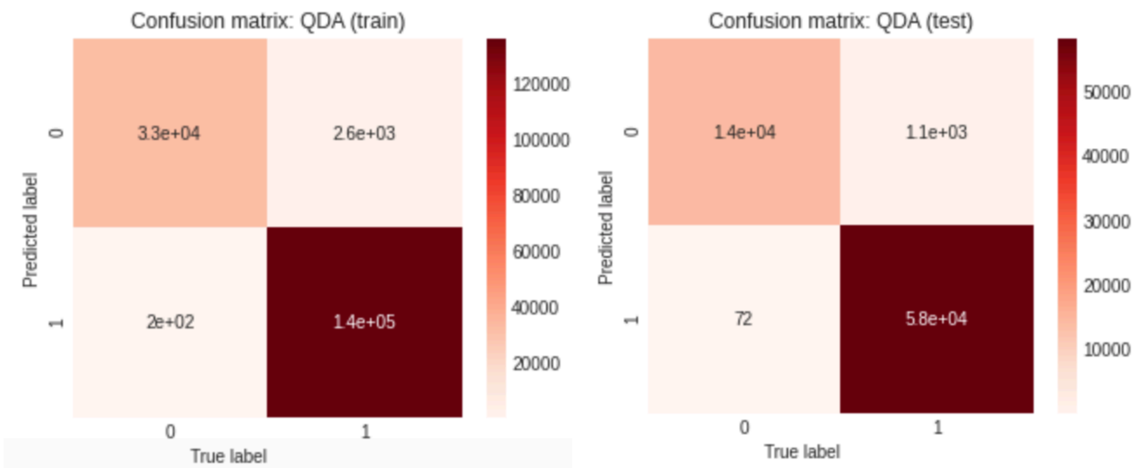

Fig. 8.3: Confusion matrix of Quadratic Discriminant Analysis

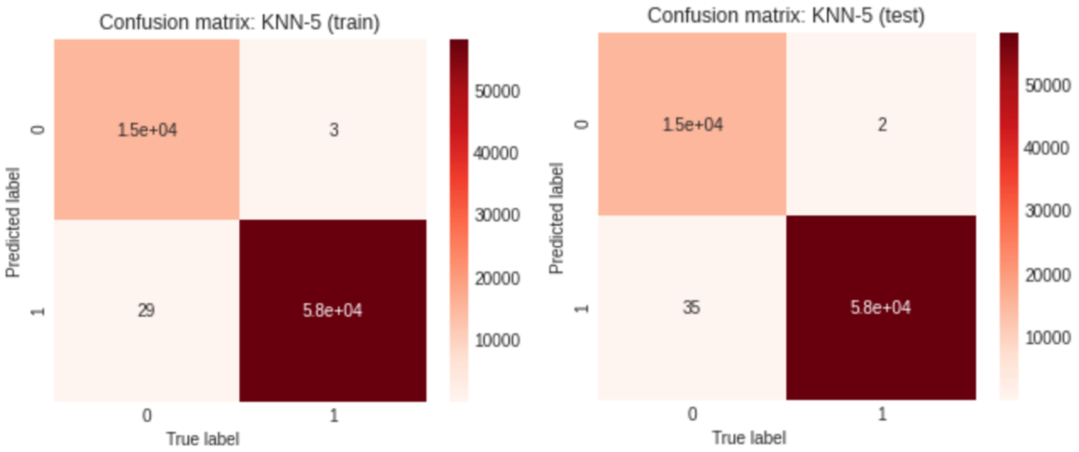

Fig. 8.4: Confusion matrix of K-Nearest Neighbor (K=5)

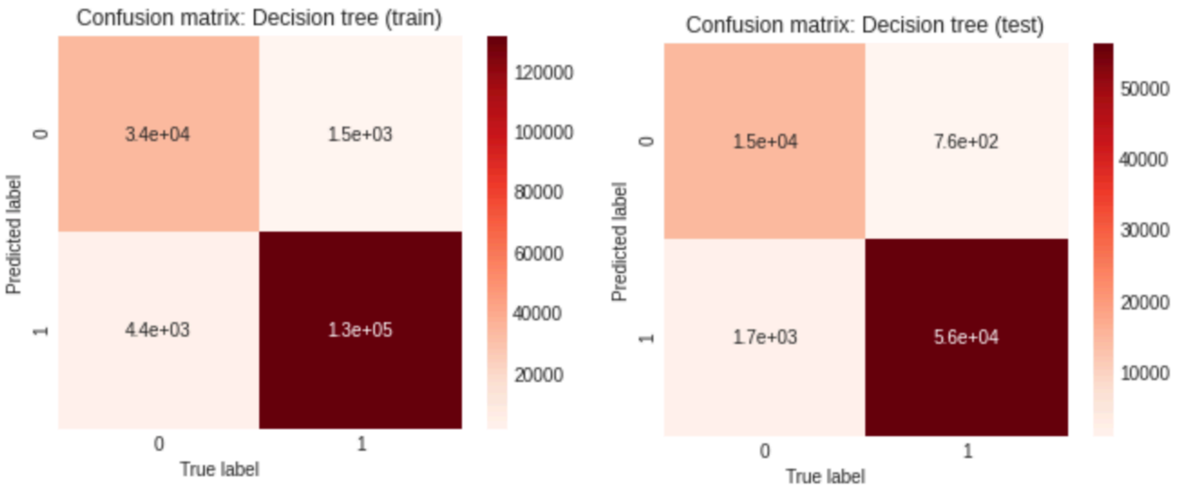

Fig. 8.5: Confusion matrix of Decision Tree

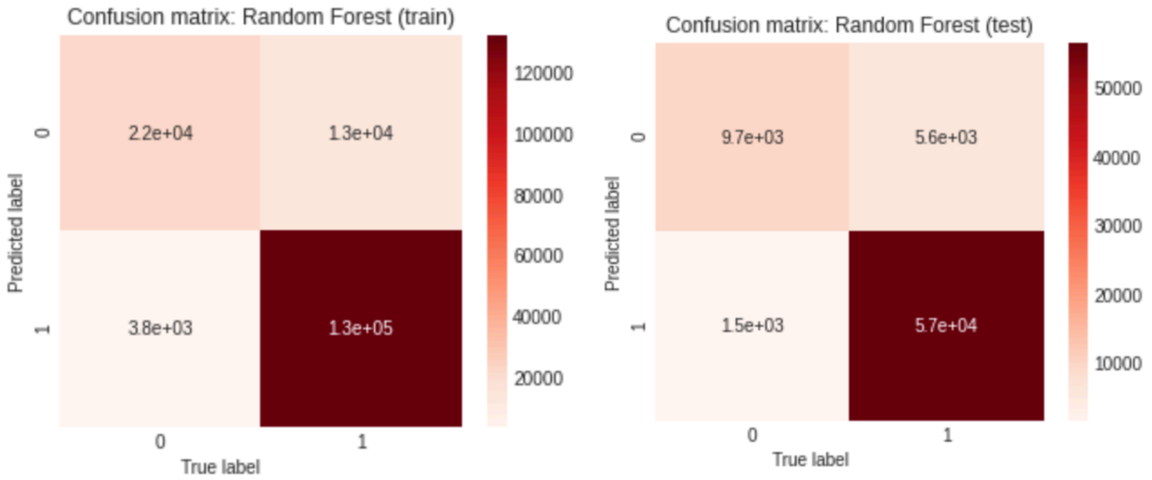

Fig. 8.6: Confusion matrix of Random Forest

After fine-tuning mentioned parameters in part 8, we obtained the score of every model. Table 8.3 compares the scores before and after fine-tuning. As you can see, some models after fine-tuning perform better and these models are more suggested to apply. Although, KNN performance seems better than other models, Decision Tree and Random Forest have more reliable results. So, I suggest applying Decision Tree or Random Forest to classify the skin segmentation dataset.

	Logistic Reg.	LDA	QDA	KNN	Decision Tree	Random Forest
Before	0.9052885	0.93265594	0.98399031	0.99949672	0.965994722	0.902894529
After	0.93586406	0.93153038	0.97630777	0.992511983	0.97626695	0.9763812153
improved	**YES**	**NO**	**NO**	**NO**	**YES**	**YES**

Table 8.3: Scores of the models after and before fine-tuning

Conclusion:

1. The skin segmentation dataset is unbiased
2. The features (Blue, Red, Green) are highly correlated and it is suggested to not eliminate any of the features.
3. Non-linear models perform better than linear models.
4. QDA has high performance so the data must be normal. (Gaussian Distribution)
5. Before fine-tuning, KNN with K=5 has the best score among other models.
6. Random Forest has the most improvement by fine-tuning. (max depth)
7. After fine-tuning, KNN with K=21 has the best score among other models.