# Project in Data Science - Medical Imaging

**Tetiana Tretiak**
IT university of Copenhagen
BSc Data Science
tetr@itu.dk

**Mariia Zviahintseva**
IT University of Copenhagen
BSc Data Science
mazv@itu.dk

**Kateryna Tkachuk**
IT university of Copenhagen
BSc Data Science
ktka@itu.dk

## Abstract

This project focuses on using image processing techniques to automatically classify skin lesions. By leveraging a specific dataset, the PAD-UFES-20, we firstly identified important characteristics of skin lesions which inform the selection of the most relevant features for automated classification. An annotation guide was created to ensure consistent and accurate evaluation of these features throughout the project.

Classifiers were then developed and trained using these selected features, tested for accuracy and reliability against new images. The aim is to demonstrate how machine learning can significantly enhance the accuracy and efficiency of medical diagnostics, specifically by automating the detection of skin lesions, making it a more accessible and practical tool in clinical settings.

## 1 Introduction

### Relevance and issues of the topic

Skin cancer represents a significant public health concern worldwide, and this disease is steadily rising over recent years. The number of skin cancers exceeds the number of all other cancers combined. There were more than 150,000 new cases of melanoma of skin in 2020.

Early detection plays an essential role in mitigating these consequences and improving treatment outcomes. However, the accurate diagnosis of skin cancer remains a complex challenge, requiring careful evaluation of various clinical and dermatoscopic features. Automating aspects of this diagnostic process through computer-based classifiers holds huge potential in enhancing the efficiency and accuracy of skin cancer detection.

### Provided info and data set

For our project, we have access to the PAD-UFES-20 dataset, which consists of 2,298 samples of six different types of skin lesions categorized into three skin cancers (BCC, MEL, SCC) and three skin diseases (ACK, NEV, SEK). Each sample consists of a clinical image and up to 22 clinical features including the patient's age, skin lesion location, Fitzpatrick skin type(colour), and skin lesion diameter. The images present in the dataset have different sizes because they are collected using different smartphone devices, but all are stored in .png format. The features are available in a CSV document in which each line represents a skin lesion and each column - a metadata feature.

### The main purpose

The primary purpose of this study is:

- to see if we can turn the ABCD rating system used by doctors into computer programs;
- to help improve how we diagnose skin cancer by using classifiers, which could help find it earlier and make treatment more effective for patients;
- to see if it is possible to make computer programs better at detecting skin cancer by looking at other details that doctors might not usually consider.

By doing this, we would find out what computer-based methods are good at and where they might struggle in diagnosing skin lesions.

## 2 Related Works

While preparing and investigating for this project, we referred to several sources, including:

- **Statistical Image Processing for Skin Lesion Detection (1, )** In this study, researchers presented development of an automated system that can help dermatologists diagnose skin lesions more accurately using statistical image processing techniques.

The authors aimed to detect specific dermoscopic features in images of skin lesions, including asymmetry, color variability, dots, and globules. To achieve this, they employed various image processing techniques such as segmentation, color clustering, and feature extraction, combined with statistical analysis. Their work highlights the importance of employing advanced algorithms and statistical methods in processing dermoscopic images, enabling the detection of crucial diagnostic features.

In the case of our project on medical imagining, mentioned study provided a strong foundation for our approach to processing and analyzing images.

- **Diagnosis of Skin Lesions Using Dermoscopic Images and Image Processing Techniques (2, )**

In a book chapter titled ”Diagnosis of Skin Lesions Based on Dermoscopic Images Using Image Processing Techniques” by Ihab Zaqout (2019), the author delves into the utilization of image processing techniques for the diagnosis of skin lesions. The chapter covers various aspects of this research area, including image pre-processing, feature extraction, segmentation, and classification. Techniques such as thresholding, edge detection, morphological operations, and color-based segmentation were described. Additionally, the chapter discusses the importance of feature extraction and selection for the identification of relevant characteristics of the lesions, such as shape, size, texture, and color. Based on these features, differentiation between benign and malignant skin lesions can be determined.

Zaqout’s work provides a comprehensive overview of the different image processing techniques used for skin lesion diagnosis from dermoscopic images. In the context of our project, Zaqout’s chapter provided valuable insights into various image processing techniques and classification methods that can be applied to improve the accuracy and reliability of our work.

## 3 Methodology

### 3.1 Data exploration

After processing all 2,298 images with all the information about them, namely established diagnoses, the following data distribution was revealed:

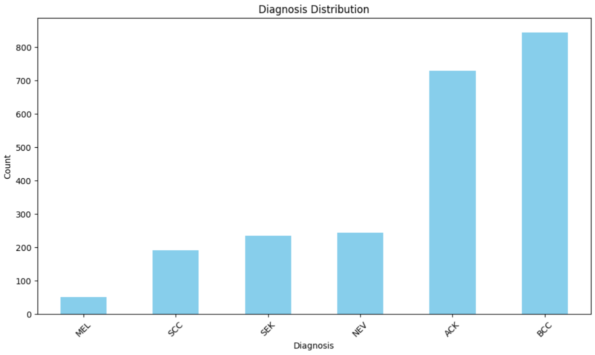


Figure 1: Distribution of 6 diagnosis

Upon initial observation, the distribution of diagnoses appears somewhat uneven. One may say that there are more data with cancer, since the diagnosis of BCC is in the lead. However, a closer examination exposes a more balanced distribution:

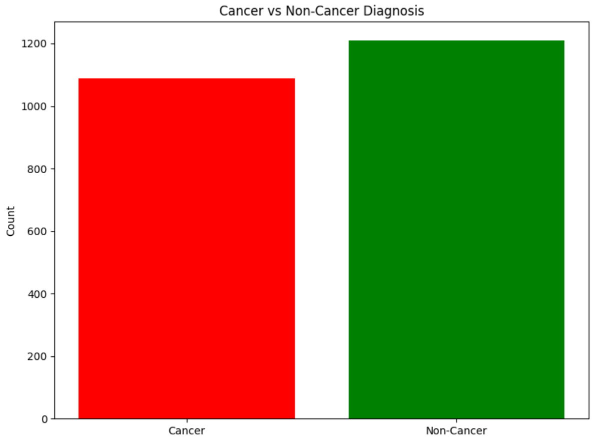


Figure 2: Cancer vs. non-cancer distribution

When categorizing all diagnoses into two groups — cancerous and just skin diseases — we notice that the data distribution is almost even. To be precise, the combined count of cancerous diagnoses, including BCC, SCC, and MEL, is slightly less than that of skin diseases - ACK, NEV, SEK. This insight underscores the importance of categorizing and analyzing data to uncover nuanced patterns that may not be immediately apparent.

## 3.2 Data Prepocessing

### 3.2.1 Data cleaning

Before we started analyzing the data, we needed to clean it up. This means getting rid of any mistakes or irrelevant information that could confuse our analysis.

First of all, we checked the data for any duplicate images to ensure that each image is unique and contributes meaningfully to our analysis. Then we checked any missing or corrupted images, and either replaced them with suitable alternatives or excluded them from the dataset.

By thoroughly cleaning the data, we ensured that our analysis is based on high-quality, reliable information, eventually leading to more accurate results.

### 3.2.2 Data collection

The next step was gathering 126 skin lesion images which involved selecting a diverse and representative sample of images with various types of lesions, skin regions, and other features. A comprehensive dataset was a foundation for our research and analysis. So we could effectively train and evaluate our computer-based diagnostic algorithms, improving the ability to detect and classify skin lesions more accurately.

## 3.3 Feature selection

In our project, we have selected a set of relevant dermascopic features based on their significance in determining whether a lesion is benign or malignant. Additionally, we considered the features present on our group images from given dataset. These features include color variability, dots and globules, asymmetry, and compactness (border irregularity).

- **Color Variability**

Color variability is a significant indicator in the assessment of skin lesions, as it can reveal information about the distribution of melanin and hemoglobin within the lesion. This feature was also utilized in the work of De Vita et al. (2011)(1, ), demonstrating its importance in statistical image processing for skin lesion detection. Furthermore, color variability is one of the criteria included in the three-point checklist (Argenziano et al., 1998(3, )), which is a well-established dermoscopic diagnostic algorithm.

- **Dots and Globules**

Dots and globules are known to be crucial diagnostic features in dermoscopy (Braun et al., 2002(3, )). They represent localized structures within the lesion and can provide insights into the underlying pathological process. Detecting dots and globules was a central aspect of the study by De Vita et al. (2011)(1, ), highlighting their importance in automated melanoma detection systems.

- **Asymmetry**

Asymmetry is a key criterion for identifying malignant skin lesions (Pehamberger et al., 1987(6, )). In clinical practice, dermatologists evaluate the asymmetry of a lesion to determine its potential malignancy. Moreover, asymmetry is an essential component of both the ABCD rule (Nachbar et al., 1994(5, )) and the three-point checklist (Argenziano et al., 1998 (3, )).

- **Compactness (Border Irregularity)**

Compactness, or border irregularity, is another significant feature for characterizing skin lesions. Lesions with irregular borders are more likely to be malignant (Friedman et al., 1985(4, )).

## 3.4 Image annotation

In order to focus on the medically relevant parts of the skin, our team manually created segmentation masks using Label Studio. This helped to isolate the skin lesion from the surrounding skin, ensuring that only isolating the critical areas for observation are considered during the analysis.

In a second stage of our project, to , firstly, manually access mentioned features on skin lesion we created a annotation guide that outlines specific criteria for assigning scores to each feature:

- **Asymmetry**
    1. Examine the lesion for horizontal and vertical symmetry.
    2. Assign a score based on the following criteria:
        - 1: If both horizontal and vertical symmetry are present.
        - 2: If at least one of them is present.
        - 3: If symmetry is absent.
- **Color**
    1. Investigate the presence of various colors within the lesion, including white, red, light brown, dark brown, blue-gray, and black.

2. Assign a score ranging from 0 to 6 depending on the number of colors identified.

- **Dots and Globules**

1. Identify dots and globules using the following characteristics defined by Kittler et al. (2016a)(7, ):
   - Small, round, and well-circumscribed structures.
   - Color variations, including brown, black, gray, blue, and red.
2. Assign a score of 1 if either dots or globules are present, and a score of 0 if they are absent.

## 3.5 Data collection

After extracting the necessary data from the dataset and manually labeling it using masks and annotations according to an annotation guide we agreed upon, we stored all data in a shared Git repository to facilitate easy access for future model development.

We thoroughly recorded image IDs, annotator IDs, and annotations in a CSV file, ensuring at least two annotators reviewed each image to reduce bias. We organized images assigned to our group and their corresponding masks into a separate folders for easy access.

Furthermore , we calculated Krippendorff's alpha for our manual annotations to estimate the agreement score within the group.

In the future, we will use these data to evaluate our classifier. Additionally, we obtained masks and images from other groups, storing them in separate folders for use in training the classifier.

We also extracted patient information and diagnoses from the metadata to serve as labels for the classification algorithm.

## 3.6 Feature extraction

### 3.6.1 Asymmetry

To easily determine the symmetry of the object in the photo we used binary masks obtained after the segmentation of the images. The feature performs image symmetry analysis using the OpenCV library in Python.

The main function calculates the asymmetry ratio of a given image in such a way. It first finds the center of mass of the image, and then determines the maximum distance from this center to the image's edge. Next, it crops the image to a square region around the defect, ensuring that it's centered on the center of mass. After that, the symmetry of the images is checked along four axes: vertical, horizontal, and two diagonals. Finally, it calculates the ratio of asymmetry based on the absolute difference between the image and its reflected counterpart along each axis.

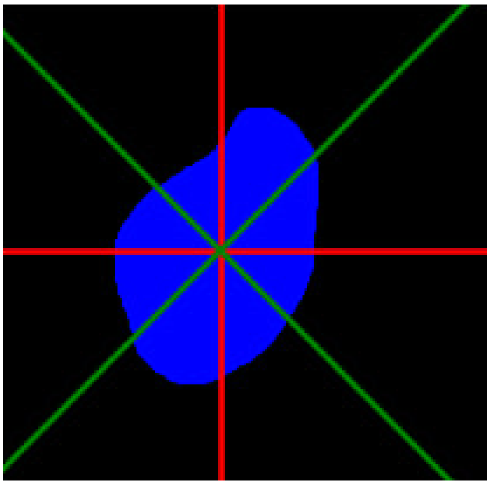


Figure 3: Rotation

One extra function categorizes the symmetry level based on the asymmetry ratio:

- If the ratio is less than 0.1, it's considered symmetric(1).
- If it's between 0.1 and 0.3, it's considered partly symmetric(2).
- Otherwise, it's considered asymmetric(3).

For a perfect understanding of which figure is asymmetrical, we used a fully asymmetrical object:

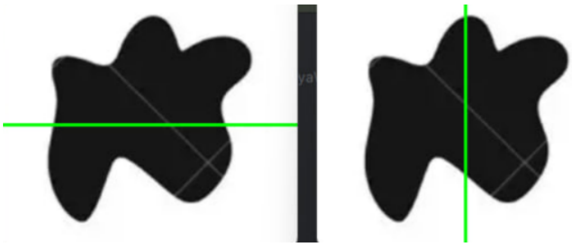


Figure 4: Asymmetry

Whereas to demonstrate complete symmetry, the circle was used:

In contrast, we have also tried a feature that neither centers the object in the image nor rotates it,

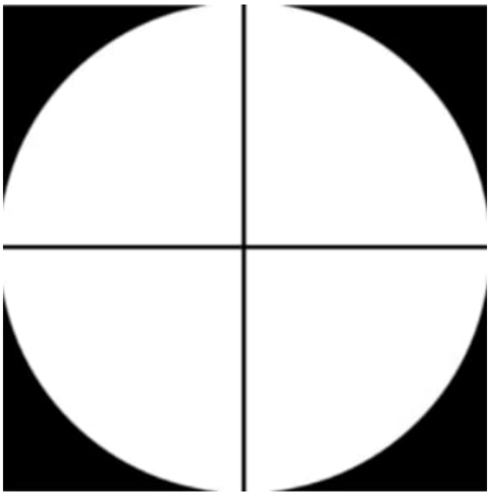


Figure 5: Symmetry

so instead of 4 diagonals, we had only 2, which in theory could provide us with worse accuracy of symmetry. However, after testing, it was found that it performs 7 times better than the one without the rotation function.

Nevertheless, we did not use this feature to our classifier, as it does not include all the necessary functionality to determine symmetry, although it gives a better result.

### 3.6.2 Colors

To analyze the presence and number of colors in a skin lesion image, a color segmentation process was implemented using binary masks obtained after lesion segmentation. For this stage we created code on Python using such libriries as numpy and theOpenCV library to analyze the colors present in the lesion. The input image is first converted from the RGB color space to HSV (Hue, Saturation, Value) color space using the cv2.cvtColor function, which enables color-based segmentation.

Next, predefined HSV color ranges are specified in the dictionary, which includes colors commonly found in skin lesions: white, red, light brown, dark brown, blue-gray, and black. These color ranges define the lower and upper bounds of HSV values for each color, allowing the algorithm to identify specific colors in the lesion image.

For each color, a mask is created using the cv2.inRange function, applying the specified HSV color range. The provided binary mask, which isolates the lesion from the surrounding skin, is then applied to the color mask using the cv2.bitwise and operation. This ensures that only pixels falling within the lesion area and the specified color range are considered. Contours, or regions with similar pixelels, are identified within the masked color mask using the cv2.findContours function. Small contours are filtered out using the cv2.contourArea function, eliminating noise and focusing on significant color regions.

If at least one significant contour is found for a given color, the contours are stored in the s dictionary, and the color is added to the present colors list. The function returns the number of segmented colors found in the lesion. Firstly, to test our color segmentation algorithm we used a simple ”toy” image (Figure 6), which features six distinct colors commonly found in skin lesions: red, white, black, blue-gray, dark brown, and light brown.

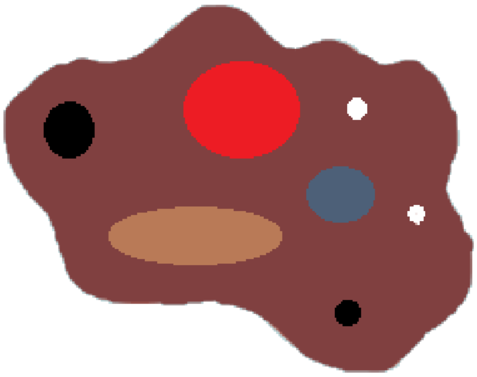


Figure 6: ’Toy’ color image

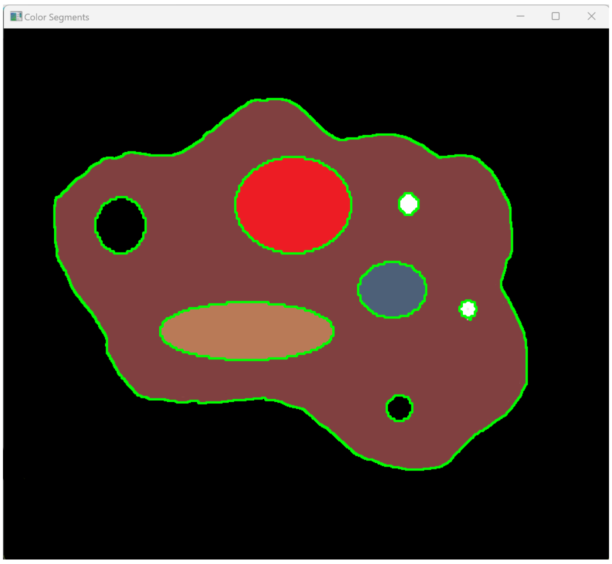


Figure 7: Processed ’toy’ image

As seen in Figure 7, our algorithm successfully identified six color regions within the test image, proving its ability to segment skin lesion images based on the predefined color ranges.

#### 3.6.3 Dots and globules

To determine the presence of circular shapes in a skin lesion image, we created python code using OpenCV and NumPy libraries for shape analysis in the lesion images.

The process begins by pre-processing the input image to improve contrast and reduce noise. CLAHE (Contrast Limited Adaptive Histogram Equalization) and median blur techniques are applied for this purpose. Then, adaptive thresholding binarizes the image, enabling shape segmentation from the background.

Next, morphological operations are performed to remove small noises and improve the detection of larger objects. Contours, or regions with similar pixels, are identified within the image using the cv2.findContours function. Contours that are too small or too large are filtered out to focus on detecting dots and globules of relevant sizes.

For each remaining contour, the algorithm checks if it is approximately circular and does not touch the image border. If a contour passes the circularity and border checks, it is considered a valid dot or globule returning one , in opposite case zero is returned.

To test our dots and globule detection, we also used ”toy” image featuring various circles of different sizes, simulating dots and globules commonly found in skin lesions(Figure 8).

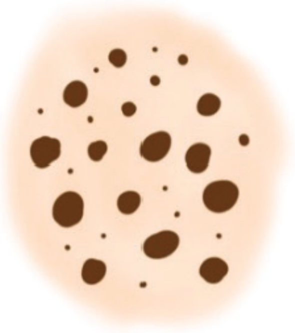


Figure 8: Toy image for dots and globules

The algorithm successfully identified the circular shapes within the test image(Figure 9).

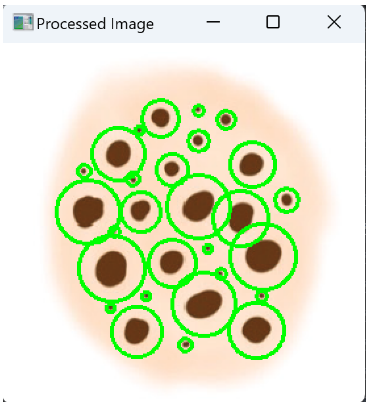


Figure 9: Processed image with dots and globules

#### 3.6.4 Compactness

To assess the compactness of skin lesions, we developed a code on python which involves the OpenCV and NumPy libraries.

Firstly, the input image undergoes preprocessing: contrast enhancement using Contrast Limited Adaptive Histogram Equalization (CLAHE) and noise reduction with median blur.

Then, adaptive thresholding is applied to binarize the input image, creating a binary mask for lesion segmentation. Next, morphological operations remove small artifacts and enhance the detection of larger shapes within the lesion.

The cv2.findContours function is then used to identify contours, or regions with similar pixel values, within the image. From these contours, the largest one is selected, as it typically corresponds to the primary lesion region. The area and perimeter of this contour are calculated using OpenCV’s cv2.contourArea and cv2.arcLength functions.

Finally, compactness is computed using the formula (perimeter ** 2) / (4 * pi * area).

### 3.7 Choosing and developing classifier

We chose several classifiers known for their effectiveness in binary classification tasks:

- K-Nearest Neighbors (KNN) with configurations of 1 and 5 neighbors
- Ensemble methods: Random Forest, Gradient Boosting, and AdaBoost, each with 100 estimators
- Decision Tree, Logistic Regression (standard and class-weight balanced), Stochastic Gradient Descent (SGD), and Gaussian Naive Bayes

We integrated each classifier into our analysis pipeline, which began with the extraction of key lesion features followed by the merging of these features with patient diagnostic metadata to form a complete dataset.

**Data Integration and Preparation** We extracted features such as asymmetry, colors, dots, globules, and compactness from images. We then combined these features with patient metadata, including diagnostic labels and patient IDs, to create a unified dataset for training. We prepared a feature matrix $X$ and a target vector $y$ for classification. The target vector marks skin lesions as either cancerous (BCC, MEL, SCC) or non-cancerous.

**Training and Validation** We employed a Group K-Fold cross-validation method, partitioning the data into five groups to ensure that each group served once as a test set. This method helped validate the effectiveness of each classifier and maintained the integrity of the evaluation process.

**Performance Evaluation** We measured classifier performance primarily through accuracy. Each model was serialized with Python's `pickle` module after training to make future evaluations easier. We later evaluated these models on the entire dataset, using precision, recall, and F1-score metrics to provide a detailed classification report.

**Application of Chosen Classifier** Following the performance evaluation, we selected the classifier demonstrating the highest effectiveness for practical application. The process involved utilizing the classifier to predict the probability of cancerous conditions in new, unseen data.

**Prediction and Result Compilation** We created a script that integrates the extraction of features, prediction of probabilities, and generation of a results summary. This script:

- Extracts features such as asymmetry, colors, dots and globules, and compactness from new image data.

- Merges these features with patient diagnostic metadata to ensure each prediction is linked with the correct patient ID and image ID.

- Loads the selected high-performance classifier from a serialized file using Python's `pickle` module.

- Predicts probabilities of each lesion being cancerous or non-cancerous.

- Compiles the probabilities, actual labels, and identifiers into a structured format.

**Output and Visualization** The results of these predictions are saved into an Excel file for easy access and further analysis. Additionally, this data facilitates the visualization of results, allowing us to closely examine the predictive power of the classifier and to assess its practical effectiveness in a clinical setting. The visualization is performed using `seaborn` library which helped in plotting the confusion matrix.

### 3.8 Evaluating classifier

To accurately assess the performance of our developed classifiers, we implemented a precise evaluation process. This process involved extracting features from previously unseen data, which were then integrated with patient diagnostic information to create a comprehensive test dataset.

The ultimate goal of this evaluation was to identify and select the most effective classifier for our project.

**Classifier Evaluation** Each classifier was loaded and evaluated using a custom Python function that utilized `pickle` for loading the serialized classifier models. The evaluation process involved predicting labels for the test dataset and calculating the accuracy and confusion matrix for each model. The confusion matrix provided a detailed look at the performance, showing true positives, true negatives, false positives, and false negatives, thereby offering insight into each classifier's precision and recall capabilities.

**Automated Classifier Loading and Assessment** The classifiers were stored as serialized files and were programmatically accessed and evaluated. The script listed and loaded each classifier using their file paths, then performed predictions on the test set. This automated approach ensured that all classifiers were evaluated uniformly and efficiently.

**Results Reporting** For each classifier, the results—including the accuracy confusion matrix , recall and precision — were reported. This provided a quantitative measure of each classifier's performance, allowing us to understand the strengths and weaknesses of each model in classifying skin lesions accurately.

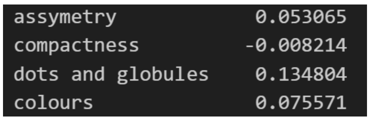
**Conclusion** This structured and systematic evaluation of classifiers helped identify the most effective models based on accuracy and confusion matrix results. The process underscored the importance of detailed testing and validation in developing reliable diagnostic tools in medical image analysis.

## 4 Analytics

### 4.1 Features analysis and their influence

To understand the impact of various features on the diagnosis, we conducted analysis using the combined dataset containing features such as asymmetry, compactness, dots and globules, colors, and diagnostic outcome.

Firstly, we calculated the correlation matrix for these features, focusing on the correlation between each feature and the diagnostic outcome. The correlation values were obtained by performing a pairwise correlation using the .corr() method from the pandas library.



```
assymetry              0.053065
compactness           -0.008214
dots and globules      0.134804
colours                0.075571
```

Figure 10: Correlation between each feature and the diagnostic outcome

- Asymmetry: The correlation coefficient between 'asymmetry' and 'diagnostic' is approximately 0.053. This suggests a weak positive correlation, indicating that as 'asymmetry' increases, the likelihood of a positive diagnosis (1 - cancerous lesion) may also slightly increase, but the relationship is not very strong.

- Compactness: The correlation coefficient between 'compactness' and 'diagnostic' is approximately -0.008. This suggests a very weak negative correlation, meaning there's almost no linear relationship between 'compactness' and 'diagnostic'.

- Dots and globules: The correlation coefficient between 'dots and globules' and 'diagnostic' is approximately 0.135. This indicates a positive correlation, suggesting that as 'dots and globules' increase, the likelihood of a positive diagnosis (1 - cancerous lesion) may also increase.

- Colours: The correlation coefficient between 'colours' and 'diagnostic' is approximately 0.076. Similar to 'asymmetry', this suggests a weak positive correlation between 'colours' and 'diagnostic'.

To further visualize the relationships between features and the diagnostic outcome, we generated violin plots(which combine combines a box plot with a density estimate to provide a more comprehensive view of the data) using the Seaborn library. These plots illustrate the distribution of each feature across different diagnostic categories.

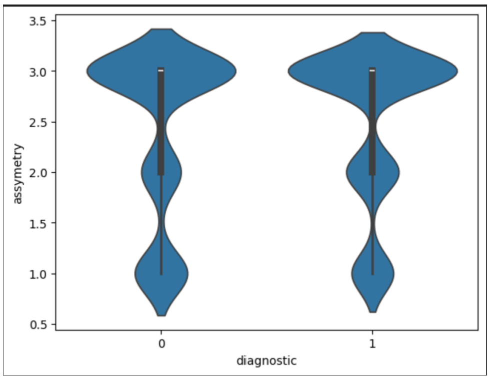


Figure 11: Analysis of asymmetry depending on diagnosis

The distribution of asymmetry appears to be similar across diagnostic categories (Figure 11). However, for the positive diagnostic category(1), there appears to be a higher concentration of points with asymmetry values of 2 and 3. In contrast, the negative diagnostic category has a higher concentration of points with an asymmetry value of 0. This observation aligns with our initial assumption about relationship between asymmetry and positive and negative diagnoses.

Similar to asymmetry, the distribution of compactness appears relatively consistent across diagnoses (Figure 12), but we can also see that within negative diagnosis violin plot shows that the density of compactness values is higher around the lower end (around 1 to 3) and gradually decreases as compactness increases. However, there are some outliers with higher compactness values. Similarly, for positive diagnosis, the density

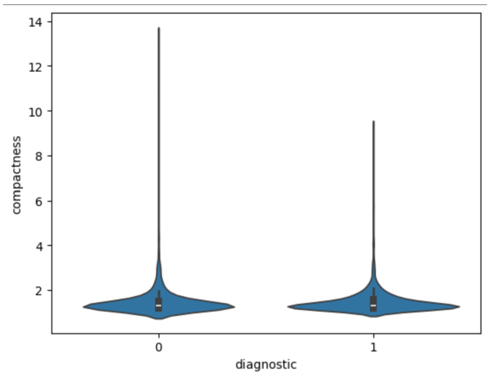


Figure 12: Analysis of compactness depending on diagnosis

is higher around the lower compactness values , but it has outliers with less higher compactness.

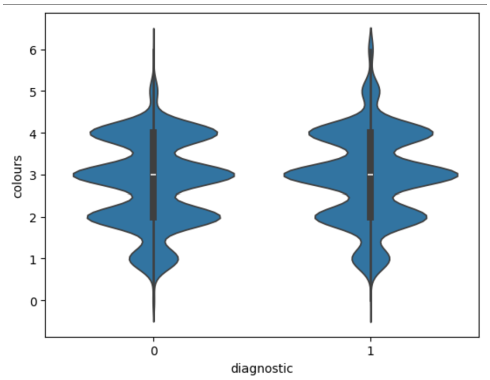


Figure 13: Analysis of colour number depending on diagnosis

The violin plots suggest a difference in the distribution of colours across diagnostic categories (Figure 14). Within the positive diagnostic category, there appears to be a higher concentration of points with colour values ranging from 3 to 6. Conversely, the negative diagnostic category exhibits a higher concentration of points with colour values 0, 1, and 2. Which also aligns with our initial assumption about relationship between colour number and diagnoses.

The violin plot shows distinct differences in the distribution of dots and globules between positive and negative diagnoses (Figure 14). There are more cases with a value of 0 in the negative diagnostic category, while the positive diagnostic category has more cases with a value of 1. This observation supports our initial assumption about the relationship between dots and globules and the diagnostic outcome.

To gain a deeper understanding of the impact

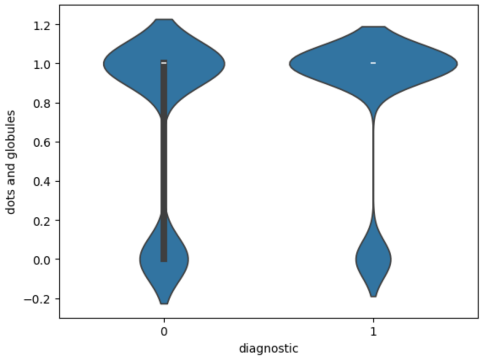


Figure 14: Analysis of dots and globules depending on diagnosis

each feature has on the classifiers' predictive performance, we trained and compared ten different classifiers on four individual features: Asymmetry, Color, Dots, and Compactness. By evaluating the accuracy of each classifier when trained on one of these features, we aimed to identify the most influential features. (Figure 15) The notation for classifier naming is as follows:

- **Classifier 1:** KNeighborsClassifier(1)
- **Classifier 2:** KNeighborsClassifier(5)
- **Classifier 3:** RandomForestClassifier
- **Classifier 4:** GradientBoostingClassifier
- **Classifier 5:** AdaBoostClassifier
- **Classifier 6:** DecisionTreeClassifier
- **Classifier 7:** Logistic Regression
- **Classifier 8:** Logistic Regression (balanced)
- **Classifier 9:** SGD Classifier
- **Classifier 10:** GaussianNB

Overall, we can see that colors number and dots and globules have better accuracy values through all classifiers, defining their significant influence compared to other features.

In addition, we will perform a feature importance analysis, which involves calculating importance values from ten different classifiers trained. This analysis focuses on the contribution of our four features on the classifiers decision-making process. Each classifier assigns importance scores to these features, representing their impact on the

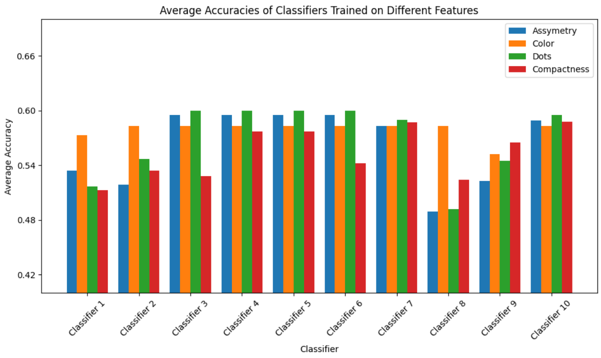


Figure 15: Accuracies of classifiers trained on different features

prediction outcome. These scores helped us understand how each feature influences the classifiers performance and enables us to refine the models for improved diagnosis prediction.(Figure 16)

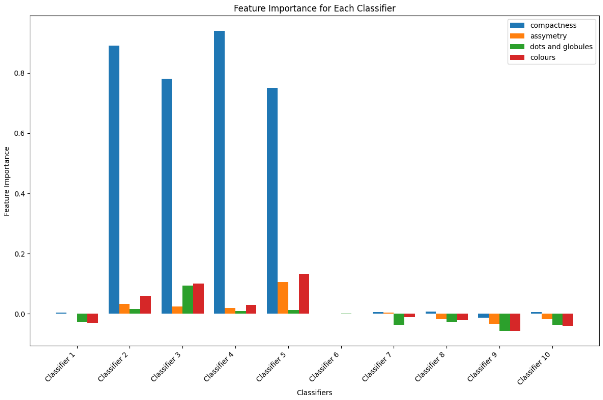


Figure 16: Feature importance for each classifier

We can see that :

1. Compactness
   - Classifier 2, 3, 4, 5, 7, and 8 assign positive importance to compactness, indicating that higher values of compactness are associated with a greater likelihood of predicting a positive diagnosis.
   - In contrast, Classifier 1, 6, 9, and 10 assign either minimal positive importance or negative importance to compactness, suggesting that its influence on predicting a positive diagnosis might be limited or inverse in certain contexts.
2. Asymmetry
   - Classifiers 2, 3, 4, 5, 7, and 8 assign positive importance to asymmetry, indicating that higher asymmetry values are generally associated with an increased probability of predicting a positive diagnosis.
   - However, Classifiers 1, 6, 9, and 10 either assign minimal positive importance or negative importance to asymmetry, suggesting a less consistent or inverse relationship with the prediction of a positive diagnosis.
3. Dots and Globules
   - Classifiers 2, 3, 4, 5, and 7 assign positive importance to dots and globules, indicating that their presence tends to contribute positively towards predicting a positive diagnosis.
   - However, Classifiers 1, 6, 8, 9, and 10 assign negative importance to dots and globules.
4. Number of colours
   - Classifiers 2, 3, 4, and 5 assign positive importance to colors, indicating that certain color patterns or intensities contribute positively towards predicting a positive diagnosis.
   - However, Classifiers 1, 6, 7, 8, 9, and 10 assign negative importance to colors, suggesting that specific color characteristics might be associated with a decreased likelihood of predicting a positive diagnosis.

## 4.2 Comparison of measures (manual and automatic)

### 4.2.1 Colours analysis

For further evaluation of colour analysis code performance, we applied it to a collection of actual skin lesion images from given dataset. Comparing its results to manual annotations using Krippendorf alpha algorithm, we got an agreement score of 0.71. This result shows that the algorithm's output matches well with manual annotations, confirming the effectiveness of code.

### 4.2.2 Dots and globules analysis

In this step we applied our code to actual skin lesion images from given dataset. By comparing its results with manual annotations using Krippendorf alpha algorithm, an agreement score of 0.51 was obtained.

While this score is not as high as those achieved for asymmetry and color assessments, it was considered acceptable given the challenges which involved misdetection of hair and skin folds as dots and globules.

### 4.2.3 Asymmetry analysis

To evaluate the performance of our feature, we set our measurements manually for the sample dataset to check how well the model results match our manual ones. This allowed us to predict how well our classifier would perform in the future.

The image-rotated model we used to determine the asymmetry gave us a result accuracy of only 10% , that is quite low. It was determined with Krippendorf implementation, which establishes the percentage of matches between the annotators´ and the model´s measurements.

Another way of resolving asymmetry was implemented neither with rotation nor centering, but only by folding photos horizontally and vertically. Even though not all the necessary functionality was used here, the result was 7 times better, which is 70% .

However, all the classifiers gave approximately the same results using both of the cases of asymmetry. Therefore, we decided to provide a method that uses more advanced functionality - asymmetry with rotation, centering, and folding.

## 4.3 Comparison of classifiers

After training the selected classifiers, we evaluated their performance based on average accuracy across five folds, as depicted in Figure 10. Among the classifiers, AdaBoost (0.591), GradientBoosting (0.589), and Logistic Regression (0.587) emerged as the top performers. Conversely, Balanced Logistic Regression displayed significantly lower accuracy of only 0.492.

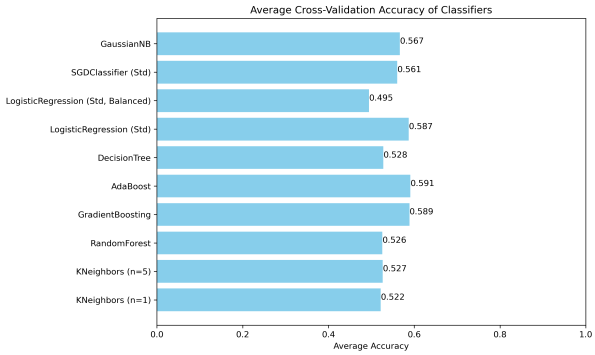


Figure 17: Average accuracies of trained classifiers

To gain deeper insights into performance, we also analyzed confusion matrices with average precision and recall rates.

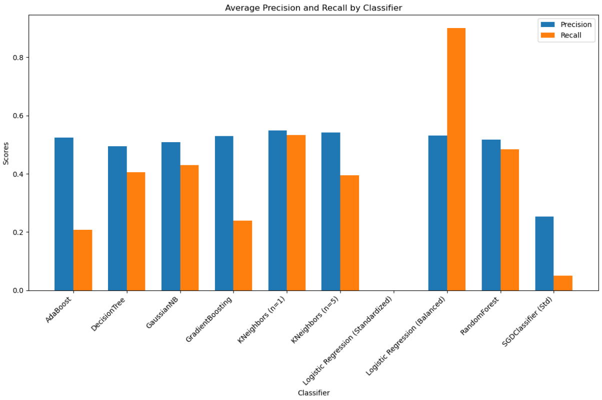


Figure 18: Average precision and recall of trained classifiers

Figure 11 shows that while Logistic Regression appeared effective based on its accuracy, it performed poorly in identifying positive cases, highlighting a critical shortfall in its application.

On the other hand, Balanced Logistic Regression, despite its lower overall accuracy, achieved a recall rate of over 0.9.

This high recall is vital in medical settings, especially in the detection of skin lesions, where failing to identify a malignant lesion could delay crucial treatment, potentially resulting in harmful effects on patient health. A high recall rate ensures that the majority of actual positive cases are identified, significantly reducing the risk of missed diagnoses.

## 4.4 Confusion matrix

In order to evaluate how well our classifier perform, we used a confusion matrix (table with 4 different combinations of predicted and actual values) where output can be two or more classes. In our case - cancer and not cancer.

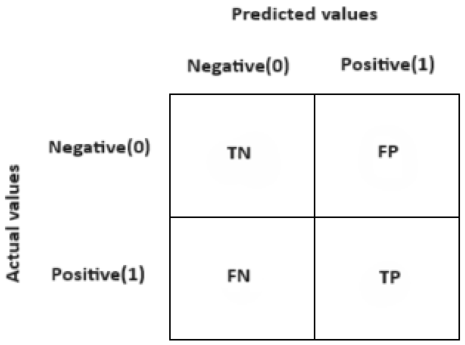


Figure 19: Confusion matrix

Here we will show how we compared classifiers and defined which one is the best. The first example is a balanced logistic regression with its output shown below:

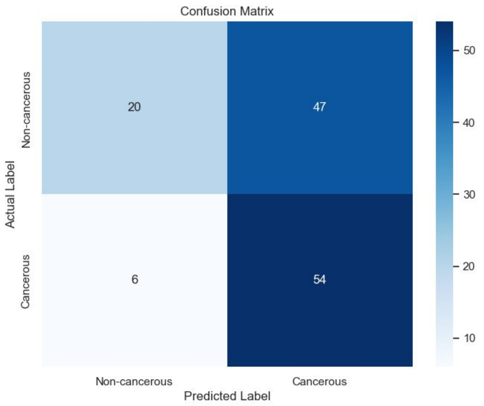


Figure 20: Balanced LR confusion matrix output

**True Positive**: Interpretation: predicted positive and it's true.

54 - the classifier predicted that there is no cancer and it actually is not.

**True Negative**:

Interpretation: predicted negative and it's true.

20 - the classifier predicted that there is a cancer and it actually is.

**False Positive**:

Interpretation: predicted positive and it's false.

47 - the classifier predicted that there is no cancer but it actually is.

**False Negative**:

Interpretation: predicted negative and it's false.

6 - the classifier predicted that there is cancer but it actually is not.

Recall, precision, and accuracy are common metrics used to evaluate the performance of classification models. All these measurements should be as high as possible. This is how we determine which classifier is the best.

**Recall** measures the ability of a model to correctly identify all relevant instances, particularly from the positive class.

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{54}{54 + 6} = 0.9$$

**Precision** measures the ability of a model to correctly identify only relevant instances, particularly from the positive predictions it made.

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{54}{54 + 47} = 0,53$$

The above equation can be explained by saying, from all the classes we have predicted as positive, how many are actually positive.

**Accuracy** measures the overall correctness of the model's predictions across all classes (positive and negative)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Accuracy = \frac{20 + 54}{20 + 47 + 6 + 54} = 0.58$$

It is difficult to compare two models with low precision and high recall (as in our example) or vice versa. So, to make them comparable, we use F-score. It helps to measure Recall and Precision at the same time. F-score uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$F - measure = \frac{2 * 0.9 * 0.53}{0.9 + 0.53} = 0.67$$

After that, we will analyze an imbalanced logistic regression in the same way, and compare the results.

All the metrics calculations:

$$Recall = \frac{0}{0 + 60} = 0$$

$$Precision = \frac{0}{0 + 0} = 0$$

$$Accuracy = \frac{0 + 67}{0 + 0 + 60 + 67} = 0.53$$

$$F - measure = \frac{2 * 0 * 0}{0 + 0} = 0$$

As a result, we can see that although the accuracy of both regressions was almost the same (58 % and 53% ), all other indicators differ significantly. An unbalanced regression returns a value of 0 everywhere (except accuracy), which leads us to the conclusion that it is quite significant to balance the classifier to get more accurate results.

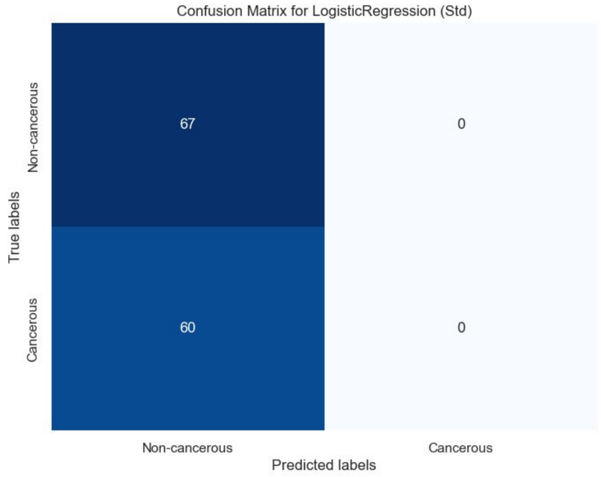


Figure 21: Imbalanced LR confusion matrix output

# 5 Results

## 5.1 Accuracy of selected classifier

Based on the detailed analysis of the classifiers, we have decided to use the Balanced Logistic Regression classifier as it demonstrated the best performance.

The performance of the Balanced Logistic Regression classifier can be understood through the following metrics, illustrated in the confusion matrix:

- **Confusion Matrix Analysis:**
  - Correctly identified 54 out of 60 cancerous cases, indicating a high true positive rate.
  - Correctly predicted 20 out of 67 non-cancerous cases, suggesting a significant area for improvement in specificity.
- **Accuracy:** The classifier achieves an overall accuracy of approximately 58%, indicating that it accurately predicts the condition of a lesion about 58% of the time across the dataset.
- **Precision:** With a precision of approximately 53%, the classifier correctly predicts cancerous lesions about 53% of the time when it classifies a lesion as cancerous.
- **Recall:** Excelling with a recall of approximately 90%, the classifier successfully identifies 90% of all actual cancerous cases, minimizing the risk of false negatives.
- **F1 Score:** The F1 score is approximately 0.67, demonstrating a balance between precision and recall, which is crucial for scenarios where detecting as many positive cases as possible is vital.

In conclusion, the Balanced Logistic Regression classifier is highly effective for detecting cancerous conditions, making it an invaluable tool in medical diagnostics.

# 6 Discussion

Addressing some encountered challenges and looking ahead to improve our study, we've identified some key things.

Firstly, we have faced the issue of class imbalance, which could significantly impact the performance of our classifiers. To counter this, we could employ a combination of resampling techniques, including both under-sampling of the majority class and over-sampling of the minority class using methods like SMOTE. Moreover, augmenting the existing data by applying transformations such as rotation, scaling, flipping, or adding noise to the images might increase the diversity of the dataset and help the model generalize better to different variations of the data. Also, the weighted loss function could be crucial for training our models effectively. With this approach, we assign higher weights to the samples from the minority class during training, essentially giving them more importance in the learning process. By doing so, we ensure that our models pay more attention to the minority class, which is often underrepresented in the dataset, and learn to distinguish it accurately.

By the way, during our research, we only used colors, symmetry, globules, and compactness for detecting cancer. But it could be possible if in the future we added to our classifier such features as the age of the person, on which part of the body the lesion was detected, whether the person has bad habits, whether this lesion is growing or is accompanied by pain sensations, they could get better results having more information to analyze.

Also, one of the reasons why our classifier can detect cancer with such a low probability is that some features are very accurate and take into account all the details, but in the end, a biased result is obtained. For example, more shades of colors are recognized, most of dark spots are recognized as globules and dots, as well as asymmetry takes into account the smallest bends.

By addressing these points and modifying our approach, we hope to improve the performance of our models and make them more reliable in detecting skin cancer accurately.

## 7 Conclusion

In this project, we focused on using machine learning to help diagnose skin cancer from patients' images. This process typically requires a lot of detail and precision from doctors and our aim was discover the way to make it easier and more accessible with the power of classification algorithms. We used a dataset called PAD-UFES-20 and tried out different ways to automatically identify features like Asymmetry, Color variability , Dots , Globules and Compactness, which are important for diagnosing skin conditions.

Our approach involved organizing and annotating the image data, selecting important features, and testing various machine learning models to find the best one. We learned how to create our own masks and labels, analyze the dataset to identify the most suitable approaches, and develop features and their measurements. We manually annotated features based on a common agreement and created scripts to automatically evaluate these features and compare the results with the manual annotations. Additionally, we explored methods to calculate the agreement score and tried to maximize the agreement between the automatically produced and manually produced scores to ensure the high performance of our scripts.

Afterwards, we also learned about different classification methods, how to preprocess and input data into them, and how to evaluate their performance using the K-Fold cross-validation method and by building a confusion matrix. This process helped us to chose the best classifier for predicting labels and their probabilities for unseen data.

We covered all the steps from extracting and preprocessing the data, to creating the classifier and loading data into it, and finally outputting the results.

We also paid a lot of attention to the analytical part, which included selecting features, choosing classifiers, and determining the best approaches for evaluating them. We chose the most reliable classifier based on this comprehensive evaluation.

We found that machine learning could really help in diagnosing skin cancer, but it also showed that we need to work more on improving how these models performance to make them as reliable as doctors. We learned that thoroughly researching documentation and understanding background information, particularly about detecting skin lesions, is crucial for developing a successful data science project which aligns with it's purpose.

Overall, this project was a great learning experience in understanding how data science can be applied to real-world problems in healthcare. We shared all our methods and findings on GitHub, which we set up according to the requirements and best data management practices. This way, we made our project accessible for others to use, evaluate and improve. While doing this we trained our skills to not only develop the project, but also being able to share it to others as a final product and make it as understandable and easy to use as possible.

## References

[1] De Vita, G. Di Leo, G. Fabbrocini, A. Paolillo, and P. Sommella. 2011. *Statistical image processing for the detection of dermoscopic* Proc. of XVIII IMEKO TC-4 Symposium

[2] Ihab Zaqout. 15 July 2019. *Diagnosis of Skin Lesions Based on Dermoscopic Images Using Image Processing Techniques.*. IntechOpen

[3] G. Argenziano, H. P. Soyer, V. Chimenti, G. Talamini, M. L. Corona, V. Sera, S. A. Binder, and R. Cerroni. 1998. *Dermoscopy of pigmented skin lesions: Results of a Consensus Meeting via the Internet*. Journal of the American Academy of Dermatology

[3] R. P. Braun, H. Rabinovitz, R. Oliviero, S. A. Kopf, and H. S. Saurat. 2002. *Dermoscopy of pigmented skin lesions*. Journal of the American Academy of Dermatology

[4] R. J. Friedman, M. B. Rigel, and D. L. Silverman. 1985. *Malignant melanoma in the 1990s: The continued importance of early detection and the role of physician examination and self-examination of the skin*. CA: A Cancer Journal for Clinicians

[5] F. Nachbar, N. Stolz, H. Merkle, U. Glaessl, K. H. D. B. Peter, H. C. Lemke, M. Staib, U. Reich, P. Stachs, and R. Berens. 1994. *The ABCD rule of dermatoscopy: High prospective value in the diagnosis of doubtful melanocytic skin lesions*. Journal of the American Academy of Dermatology

[6] H. Pehamberger, G. Binder, H. Steiner, E. M. Wolff, and G. Honigsmann. 1987. *In vivo epiluminescence microscopy: Improvement of early diagnosis of melanoma*. Journal of Investigative Dermatology

[7] Kittler, H., Pehamberger, H., Wolff, K., Binder, M., Perk, M. 2016 *Dermoscopy and digital dermoscopy for the diagnosis of pigmented skin lesions*. Dermatology Research and Practice,1-13.

[8] Noa Azaria April 7, 2024 *Exploring Feature Importance: A Comprehensive Overview and Tutorial on 7 Methods*. Aporia. https://www.aporia.com/learn/feature-importance/feature-importance-7-methods-and-a-quick-tutorial/ (Accessed: 03.05.24).

[9] Sarang Narkhede May 9, 2018 *Understanding Confusion Matrix*. Medium. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (Accessed: 03.05.24).

[10] scikit-learn *Scikit-learn: Machine Learning in Python* https://scikit-learn.org/stable/ (Accessed: 03.05.24)

[11] scikit-learn *Classifier Comparison* https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html (Accessed: 03.05.24)

[12] scikit-learn *sklearn.linear_model.LogisticRegression* https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (Accessed: 03.05.24)