Project 2 – Skin Tumor Prediction

Creation and test of various machine learning classification models on a image dataset

Alessandro Sciorilli

The objective of this project is to develop and test various machine-learning models for accurately detecting and classifying different types of skin tumors using a collection of multi-source dermatoscopic images of pigmented lesions. As per the dataset's creator, the dermatoscopic images are gathered from diverse populations and are acquired and stored through different modalities such as histopathology, follow-up examination, expert consensus, and in-vivo confocal microscopy. The dataset comprises a total of 10,015 images. Creating an effective machine learning classification system for skin tumor detection can provide substantial benefits for several reasons. Firstly, it enables early detection and diagnosis, significantly enhancing the chances of successful treatment and improved patient outcomes. By proficiently categorizing pigmented lesions from simple images, the algorithm can play a vital role in the initial stages of diagnosis without requiring immediate clinical analysis. Furthermore, a precise and efficient algorithm can aid dermatologists in reducing the risks of misdiagnosis and facilitating the comprehensive detection of skin tumors on a larger scale.

The dataset contains images from six different typologies of skin tumors:

- **Melanocytic Nevi (nv)** - Commonly known as moles, are harmless pigmented skin spots, not indicative of cancer.
- **Melanoma (mel)** - A serious skin cancer that originates from melanocytes, the cells producing skin pigment, and can spread rapidly if untreated.
- **Benign Keratosis-Like Lesions (bkl)** - Non-cancerous growths that resemble warts or keratosis and do not pose cancer risks.
- **Basal Cell Carcinoma (bcc)** - Slow-growing skin cancer that begins in the skin's basal cells, often requiring removal but having a high cure rate.
- **Actinic Keratoses and Intraepithelial Carcinoma (akiec)** - Pre-cancerous condition that has the potential to develop into invasive squamous cell carcinoma if not treated.
- **Vascular Lesions (vasc)** - Encompass various non-cancerous irregularities in blood vessels, such as birthmarks or hemangiomas.
- **Dermatofibroma (df)** - Benign skin growths resulting from minor injuries and do not involve cancerous changes.

To extract features from images, I utilize a combination of the following four image-processing machine learning models: **Mean Pixel Value, Horizontal Edges, Vertical Edges and Local Binary Patterns (LBP)**. The Machine-Learning models tested for classification are: **K-Nearest Neighbors (KNN), Naive Bayes, Support Vector Machines, Decision Tree, Random Forest and Logistic Regression.**

I download and store all the images within a folder on my desktop, alongside a **CSV file** containing the name and class of each image. In my **Jupyter Notebook**, after importing all the necessary libraries, I define the variable **'mypath'** to hold the path to the directory where my skin tumor images are located. Additionally, I read the CSV file (saved in the same directory) into a pandas dataframe and display the initial and final rows to explore the data. I exclude the columns **'lesion_id', 'dx_type', 'age', 'sex'**, and **'localization'** since they are irrelevant for the image classification process. Confirming the absence of missing data in my dataset, I proceed to read and visualize a sample image of a **Melanocytic Nevi (nv)**. Using the **'image.shape'** attribute, I determine that the image's dimensions are **450x600 pixels** and it comprises three layers: **Red, Green, and Blue.** Each layer is stored as a separate array, representing one of the colors. Subsequently, I visualize the red channel (first array), the green channel (second array), and the blue channel (third array) of the image.

While visualizing the feature distribution, I observed that the class **Melanocytic Nevi (nv)** is significantly larger than the other classes. To mitigate potential biases in classification, I performed **dataset resizing**. I randomly selected **100** images from each category to ensure a balanced representation and avoid skewed results.

Since the objective is performing a **binary classification**, I divide the primary dataframe into **six smaller dataframes**. This division allows to individually contrast Melanocytic nevi (nv) against each of the other distinct categories. Subsequently, I create a **dictionary** named **'categories'** in which each dataframe is assigned a corresponding label. For every comparison specified in the **'categories'** dictionary, I utilize a **for loop** to iterate through the corresponding list of images stored in the folder on my desktop. In this process, I resize all the images to dimensions of **100x100 pixels** and apply **four distinct image processing machine learning algorithms** to extract features from these images. These extracted features are then flattened and stored inside a long vector that I call **'all_features'**. Assuming each of my images comprises 100x100 pixels and I apply four distinct types of machine learning algorithms for feature extraction (**Mean Pixel Value, Horizontal Edges, Vertical Edges and Local Binary Patterns (LBP)**, every image will be represented as vector (containing all the flattened pixels) with **40,000** columns (features). Before applying the **Local Binary Pattern** and the **Horizontal and Vertical Edges** model, I convert the images to **grayscale.** This conversion aims to enhance the performance of these models. Regarding the Local Binary Pattern, I opt for a **radius of 1** pixel and consider the **8** surrounding pixels for comparison.

In the next step, I separate the features **X (all_features)** from the **label y** and I split the dataset into training and test using a **75/25 ratio**. After applying **rescaling** to the features using the **Standardization method**, I proceed to **train** all the machine learning algorithms (K-Nearest Neighbors (KNN), Naive Bayes, Support Vector Machines with different Kernels, Decision Tree, Random Forest and Logistic Regression). I initiate **5-Fold cross-validation** for each model and visualize the average accuracy through plotted results. Moving forward, I proceed to **fit** the models on the unseen portion of the dataset and subsequently generate a **classification report** to assess the performances of each individual model.

Generating the **elbow** graph using the K-Means algorithm **(Plot 3),** I identify k=5 as the ideal number of clusters when SSE decreases most rapidly. Beyond this point, adding more clusters doesn't significantly improve the model's fit. Choosing the number of clusters at the elbow point strikes a balance between capturing meaningful patterns in the data and avoiding excessive complexity. This approach enhances the accuracy of clustering while maintaining interpretability.

I proceed by comparing the **5-Fold Cross Validation accuracy scores** of the **Support Vector Machine** across all **six** different dataframes, aiming to identify the **best-performing kernel**. On average, the **Linear Kernel** consistently achieves the highest accuracy score among the six dataframes. Consequently, I opt for the SVM with a linear kernel for further comparisons with other machine learning models. I gather the Cross Validation accuracy scores for all the other models into a pandas dataframe and compute the average score across the six comparison classes. As a result, **Random Forest Classifier (RFC)** stands out with the highest average CV accuracy of **75.11%,** indicating robust predictive capabilities. **Logistic Regression** (LogReg) also performs well, achieving an average CV accuracy of **74.00%.** In contrast, the **Decision Tree** and **K-Nearest Neighbors** models exhibit comparatively lower performance, recording average CV accuracy scores of **68.89%** and **65.89%**, respectively. To better visualize the differences, I create a bar plot of the average cross validation accuracy for the different machine learning models **(plot 4)**.

As last step, I calculate the averages for **Accuracy, Precision, Recall and F1 across all the six different dataframes**, in order to determine the best performing Machine Learning Model.

The **Random Forest Classifier (RFC)** emerges as the top performer in this evaluation. It achieves an accuracy of **0.82**, indicating that **82%** of the predictions made by the model are correct. This high accuracy suggests the model's ability to make accurate overall predictions. Moving on, **RFC** demonstrates a precision of 0.83. This means that when **RFC** predicts a positive outcome, it is correct about **83%** of the time. Recall is 0.82, meaning the model identifies **82%** of the actual positive instances present in the dataset. F1-score is **0.82** as well, signaling that RFC achieves a good balance between precision and recall, effectively capturing positive instances while minimizing incorrect predictions. On the other end of the spectrum, the **K-Nearest Neighbors (KNN) model** appears to be the weakest performer. With an accuracy of **0.69**, precision of **0.71**, recall of **0.69**, and an F1-score of **0.67**, it falls behind the other models in terms of overall predictive performance and balance between precision and recall.

In the context of performing six binary classifications, involving **Melanocytic Nevi (nv)** versus all other tumor typologies, it becomes crucial to scrutinize the Machine Learning Model's performance across each pair of observations. Despite an overall acceptable performance, the **Random Forest Classifier** achieves an accuracy of only **0.68** when distinguishing between **Melanocytic nevi** and both **Dermatofibroma** and **Vascular lesions**. Similarly, its accuracy in accurately classifying **Melanocytic nevi versus Melanoma** stands at **74%,** indicated by an accuracy score of **0.74**. The comparison category in which the Random Forest Classifier excels is Melanocytic nevi vs. Actinic Keratoses, boasting an impressive accuracy score of **0.92**.

I plot the weighted average of the classification report metrics for each distinct model (**Plot 5**). Finally, I extract the pickles for both the **Random Forest classifier** (which is the best performer) and the scaler for deployment.

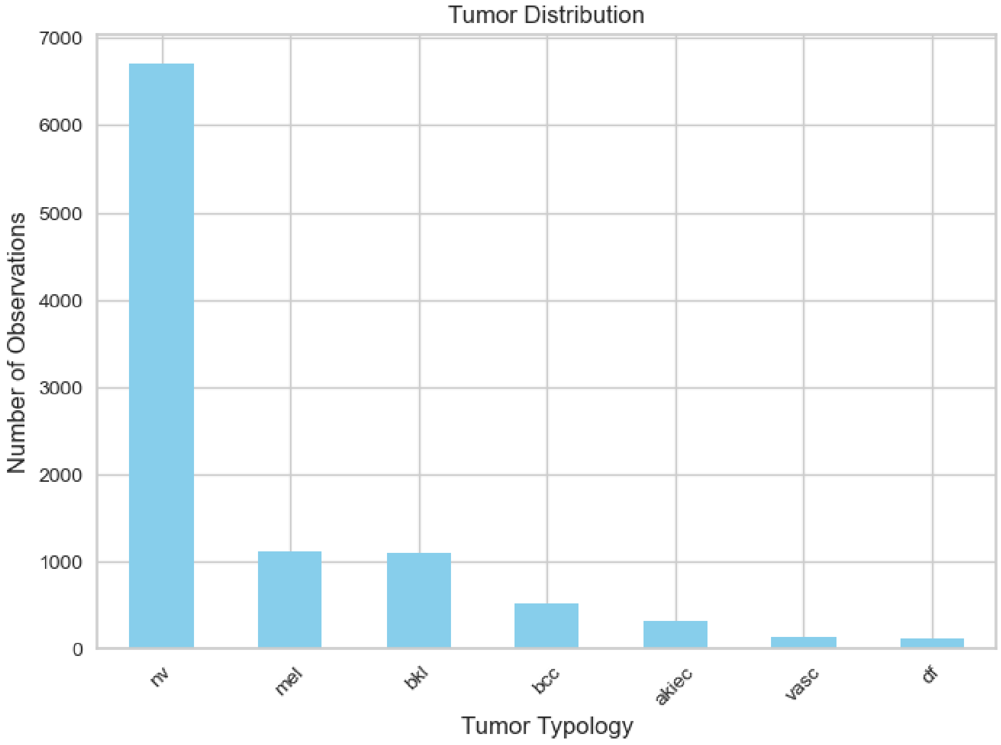
In conclusion, this project has focused on developing and evaluating machine-learning models to accurately detect and classify various types of skin tumors using a diverse collection of dermatoscopic images. For dermatologists, this model could offer an interesting tool for early detection and accurate classification of skin tumors. It complements their expertise by providing a second opinion based on thorough image analysis, reducing the likelihood of misdiagnosis and aiding in treatment planning. On the patient side, there are direct benefits in using the model for self-monitoring. Individuals can take images of their skin lesions and gain immediate insights that empower proactive self-care, enabling them to make informed decisions about whether to seek medical attention.

A potential future model development could involve focusing solely on the comparison between **Melanoma (Cancer)** and **Melanocytic Nevi (Not a Cancer).** This approach would enable training the algorithm on a larger number of observations for both classes (more than 1,000), providing an opportunity to achieve a higher level of accuracy in correctly classifying the disease. Moreover, it would allow for the distinction between a severe medical condition (cancer) and a harmless skin spot (Nevis), a differentiation that many people would find valuable and relevant.

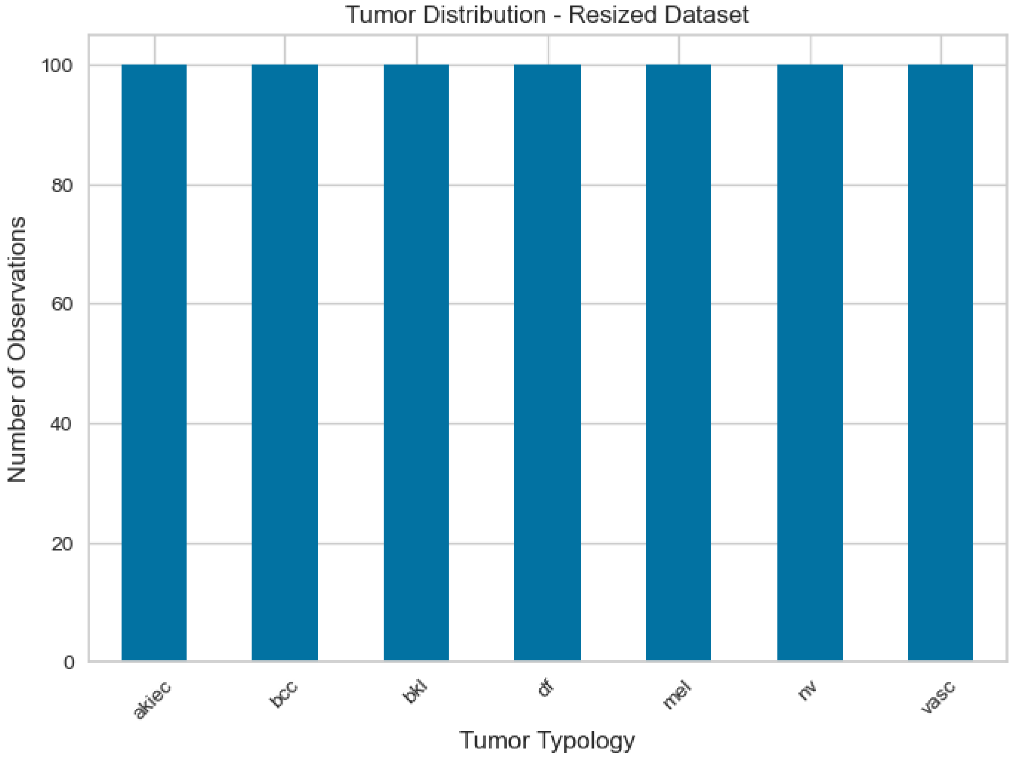
Access to the deployed model is available at the following link:
https://asciorilli.pythonanywhere.com/skin_tumor_image_classifier

The Dataset is available at the following link:
https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

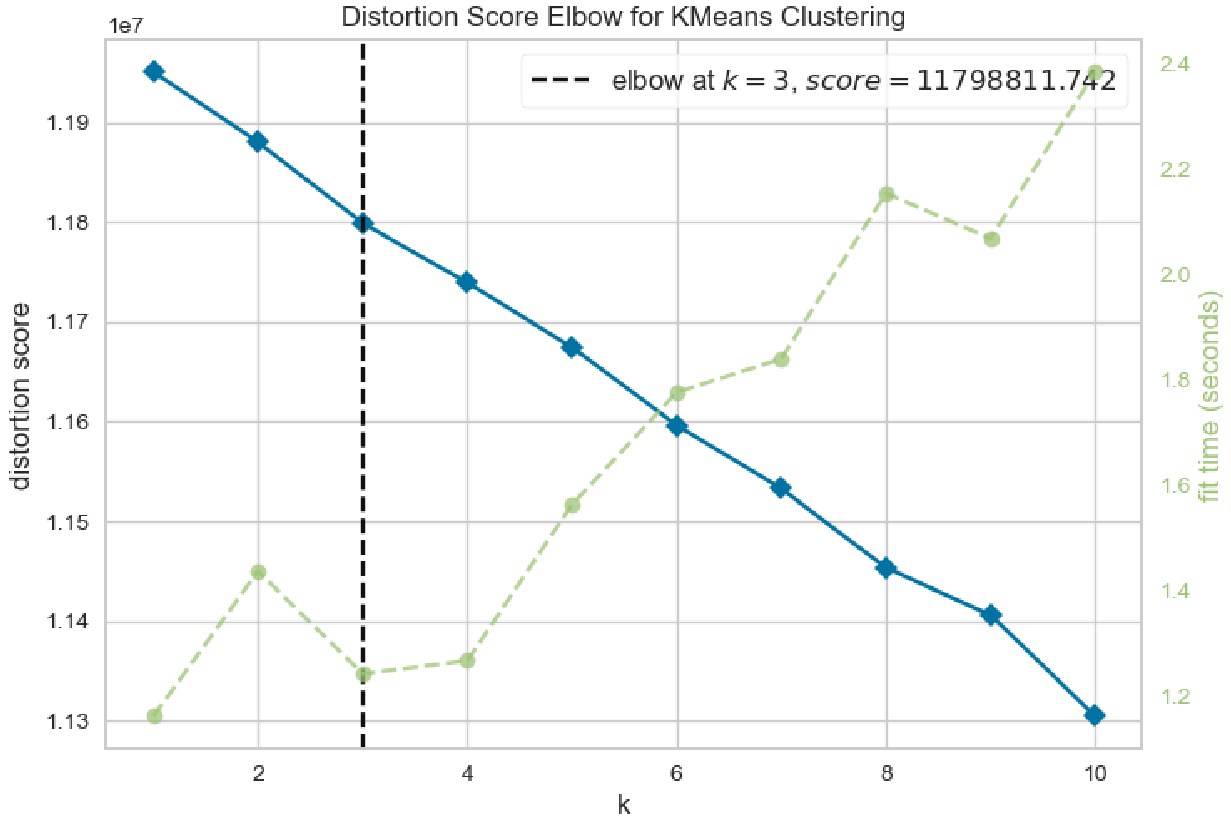
Plot 1 - Tumor Distribution



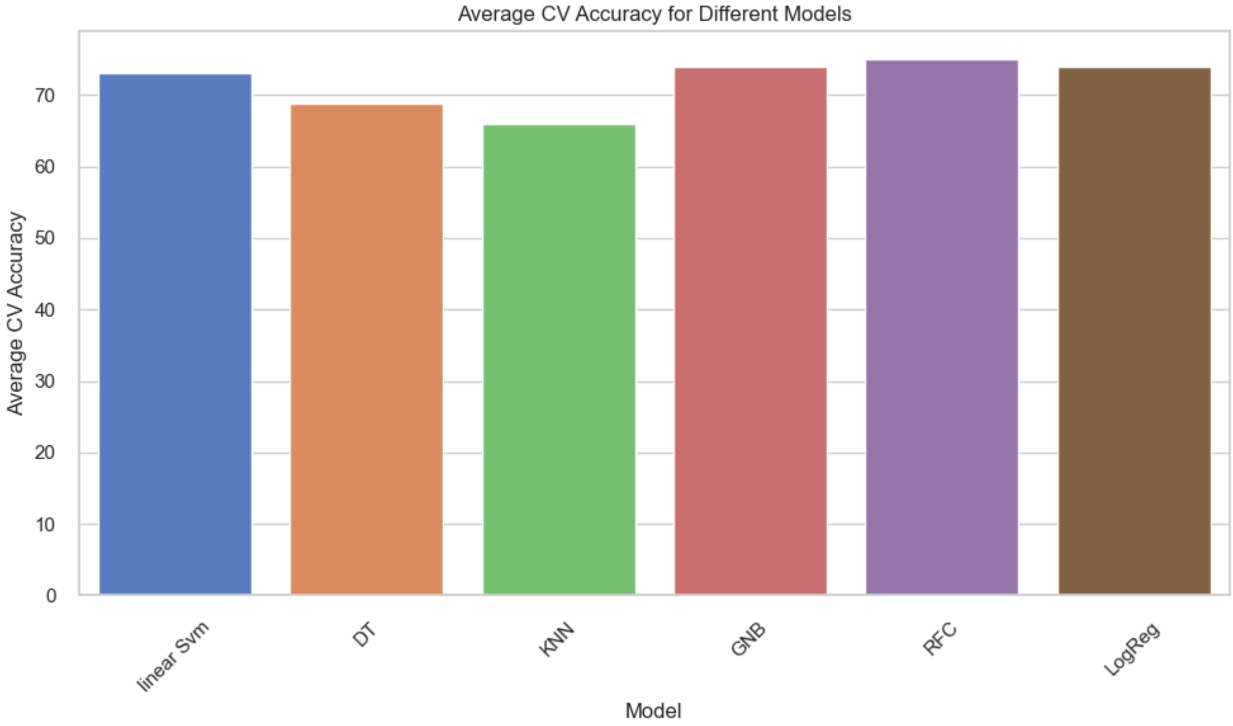
Plot 2 - Tumors Distribution with Resized Dataset



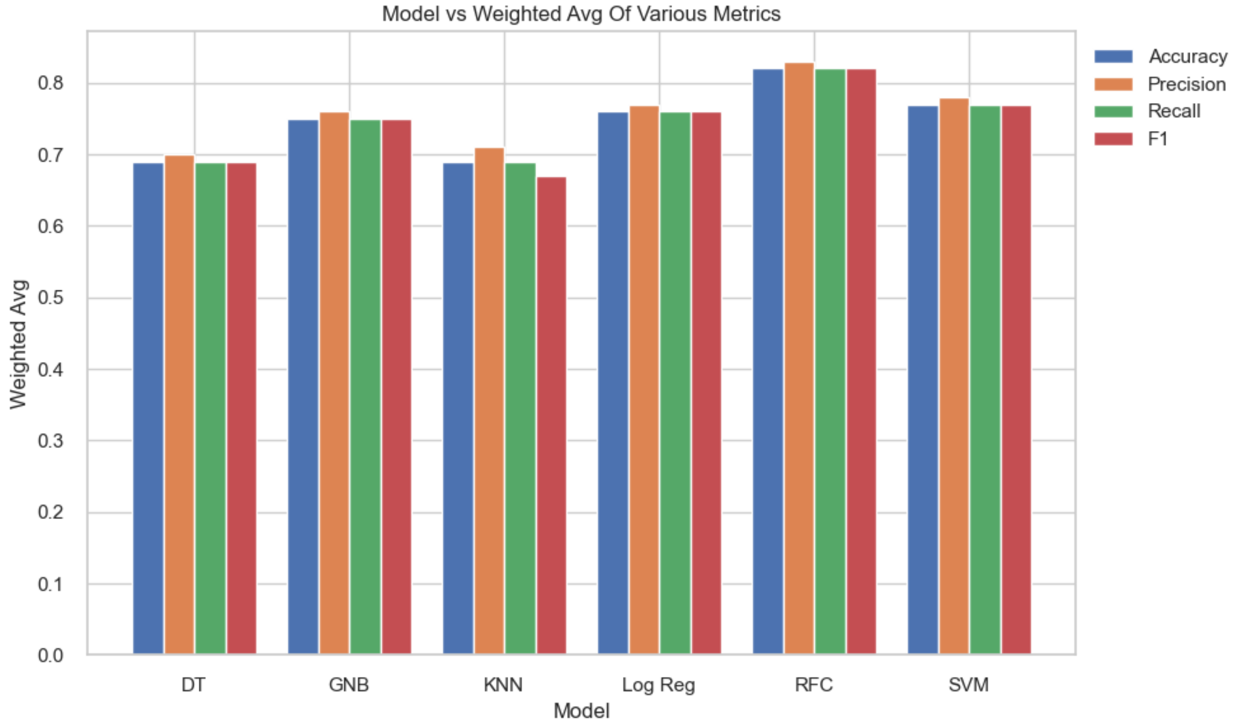
Plot 3 - Distortion Score Elbow for KMeans Clustering



Plot 4 - Average CV Accuracy for Different Models



Plot 5 - Model vs Weighted Avg Of Various Metrics


Model vs Weighted Avg Of Various Metrics
Weighted Avg
DT
GNB
KNN
Log Reg
RFC
SVM
Model
Accuracy
Precision
Recall
F1