Machine Learning Project

Skin Cancer Classification (benign or malignant)

|  |  |  |
| --- | --- | --- |
| ID | Name | E-mail |
| 213485 | Sherif Mohammed | sheif.mohammed@msa.edu.eg |
| 212473 | Yousef Amr | Yousef.amr4@msa.edu.eg |
| 212575 | Ali Yasser | Ali.yasser3@msa.edu.eg |
| 213411 | Mohamed Emad | Mohamed.emadeldin1@msa.edu.eg |

Introduction :

This is a machine learning project that aims to try different classifiers on Skin Cancer dataset (Images) and predict whether it benign or malignant

Data Description:

Skin Cancer datasets

There are two different datasets, one for the training process and one for the testing each one splitted into benign and malignant.

For preprocessing the data, we used cv2 for reading image as grayscale and resized the images to be all in the same size then flatting the 2d array for each image.

Methodology :

1. K-Nearest Neighbors (KNN): KNN is a non-parametric classifier that classifies new instances based on the majority class of its k nearest neighbors in the feature space. KNN is simple to understand and implement, and it can handle multi-class classification. However, it can be computationally expensive, especially for large datasets.
2. Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that features are conditionally independent given the class label. Naive Bayes is computationally efficient, especially with high-dimensional data. However, it makes a strong assumption of feature independence, which may not hold in some cases.
3. Support Vector Machines (SVM): SVM is a powerful classifier that finds the best hyperplane to separate different classes in a feature space. SVM can handle both linear and non-linear classification tasks by using different kernel functions. It is effective in high-dimensional spaces and can handle large datasets. SVMs are less prone to overfitting, but they can be computationally intensive.
4. Random Forest: Random Forest is an ensemble classifier that combines multiple decision trees. Each decision tree is trained on a subset of the data, and predictions are made by aggregating the predictions of individual trees. Random Forest can handle high-dimensional data, is robust to noise, and provides feature importance rankings. It is less prone to overfitting compared to a single decision tree.
5. Multilayer Perceptron (MLP): MLP is a type of artificial neural network with multiple layers of nodes (neurons). It can learn non-linear relationships in the data and is known for its ability to handle complex patterns. MLP is trained using backpropagation, and it can be used for both classification and regression tasks. However, MLP requires careful tuning of hyperparameters and can be sensitive to feature scaling.
6. Logistic Regression: Logistic Regression is a linear classifier that models the probability of a binary or multi-class outcome using a logistic function. It is interpretable and computationally efficient. Logistic Regression assumes a linear relationship between features and the logarithm of the odds ratio. It works well with linearly separable data but may struggle with complex patterns.
7. Deep Belief Networks (DBN): DBN is a type of deep learning model that combines unsupervised learning and supervised fine-tuning. It consists of multiple layers of stacked Restricted Boltzmann Machines (RBMs). DBNs can learn hierarchical representations of the data and have been successful in tasks such as image recognition and feature learning.
8. Convolutional Neural Networks (CNN): CNN is a deep learning architecture specifically designed for image analysis. CNNs are known for their ability to automatically learn and extract hierarchical features from images. They use convolutional layers to capture spatial patterns and pooling layers to reduce dimensionality. CNNs have achieved state-of-the-art performance in image classification, object detection, and image segmentation tasks.

Results and Performance Eval

Discussion and Interpretation of results:

1. Random Forest:

Before Hyperparameter tuning:

Accuracy: 0.8287878787878787

Recall: 0.87

Precision: 0.7791044776119403

F1\_score: 0.8220472440944883

Confusion Matrix:

A picture containing screenshot, text, colorfulness, rectangle

Description automatically generated

after Hyperparameter tuning :

Accuracy: 0.8318181818181818

Recall: 0.8733333333333333

Precision: 0.7820895522388059

F1\_score: 0.8251968503937007

Confusion Matrix:

A picture containing screenshot, text, colorfulness, rectangle

Description automatically generated

1. SVM

Before grid search

Accuracy: 0.7681818181818182

Recall: 0.71

Precision: 0.7634408602150538

F1\_score: 0.7357512953367875

Confusion Matrix:

[[294 66]

[ 87 213]]

after Hyperparameter tuning:

References

<https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/>

<https://michael-fuchs-python.netlify.app/2021/02/03/nn-multi-layer-perceptron-classifier-mlpclassifier/>

<https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/>

https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/