**Data Science: A Programming Approach**

**INSY 5378**

**PROJECT 1-Part 2**

**Presented by Group 3 - LIME:**

Aishwarya Prasad Venkatesh

Anurag Kumar

[Mohith Marisetti](https://uta.instructure.com/groups/12410/users/100551)

[Sangalpa Abishekapakkam Sundaram](https://uta.instructure.com/groups/12410/users/96506)

Table of Contents

Mini Project 2. Image Analysis3

Introduction and Objective3

Model Selection3

Preprocessing the data……………………………………………………………………………………………………………….…3

Predictive Models………………………………………………………………………………………………………………………….3

Conclusion……………………………………………………………………………………………………………………………………..9

References …………………………………………………………………………………………………………………………………….9

**MINI PROJECT 2. IMAGE ANALYSIS**

**2.1. Introduction and Objective:**

The main objective of this project is to build a model that identifies an object using image analysis. For the purpose of this project, as a team, we collected / clicked 80 images of four different kinds of fruits. We separated our data into train.csv and test.csv files to run the predictive models.

**2.2. Model Selection:**

We used three main predictive models for image analysis,

* + 1. SVM
    2. Random Forest
    3. Neural network classifier

**2.3. Pre-Processing the Data:**

1. Image resizing:

We used a helper function readInput to set the height and width of individual image as 100,100 respectively and converted the images into grayscale format.

2. Using Dimension Reduction:

Initially we have 10000 features per image(height\*width). To reduce the number of features we used Principal Component Analysis(PCA) and tried multiple combination of n\_componets attribute of PCA model such as5,10,20,40 respectively to test the model performance

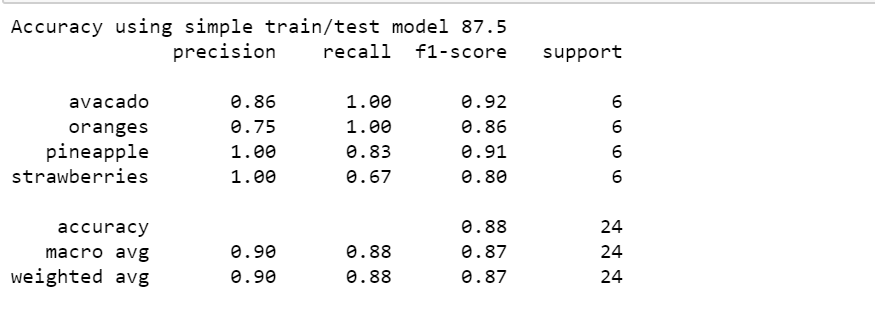
**2.4. Predictive Models:**

**2.4.1. SVM**

Using different combinations of n\_components of PCA, below table describes accuracy results for simple train, test model and model using gridsearch with cross validation approach.

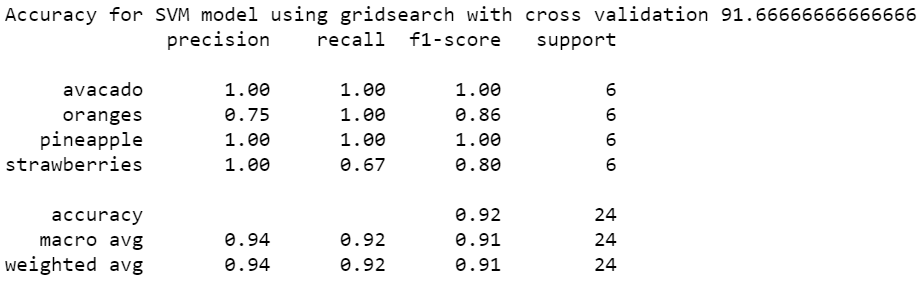
|  |  |  |
| --- | --- | --- |
| PCA(n\_components) | Simple Train/Test Model Accuracy | Model using GridSearch Accuracy |
| 5 | 75 | 83.33 |
| 10 | 91.66 | 83.33 |
| 20 | 87.5 | 66.66 |
| 40 | 75 | 75 |

The output we obtained using simple test/train model:

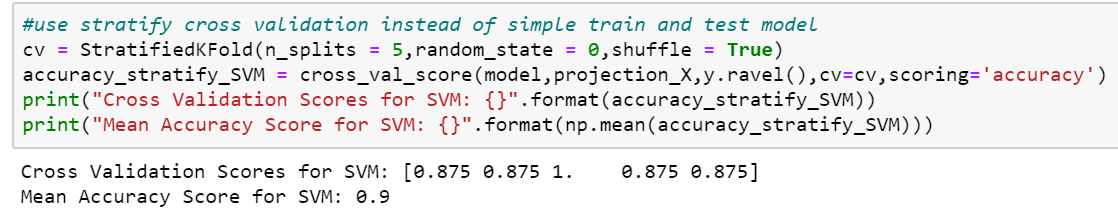


The output we obtained for SVM Classifier using GridSearch

with cross validation is



Stratify cross validation score:



ROC curve for SVM classifier:

A close up of a map

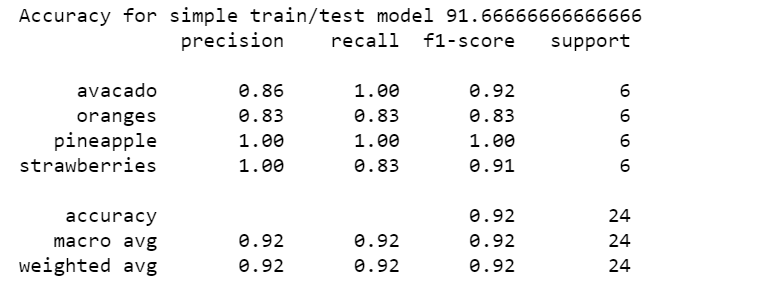
Description automatically generated

**2.4.2. Random Forest**

Using different combinations of n\_components of PCA, below table describes accuracy results for simple train, test model and model using gridsearch with cross validation approach.

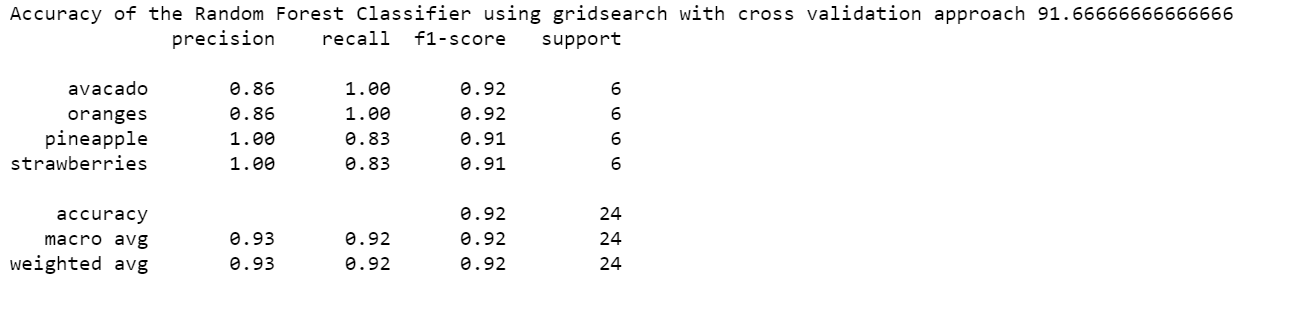
|  |  |  |
| --- | --- | --- |
| PCA(n\_components) | Simple Train/Test Model Accuracy | Model using GridSearch Accuracy |
| 5 | 79.16 | 87.5 |
| 10 | 87.5 | 87.5 |
| 20 | 91.6 | 91.6 |
| 40 | 83.3 | 79.16 |

The output we obtained using Random Forest using simple test/train model :

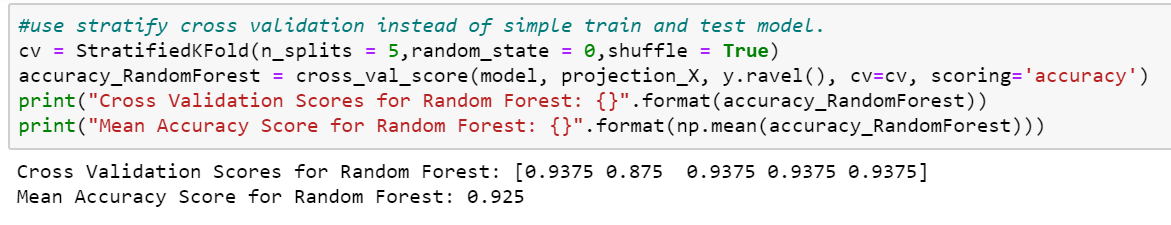


The output we obtained for Random Forest Classifier using GridSearch

with cross validation is



Stratify cross validation score:



ROC curve for Random Forest Classifier:

A close up of a map

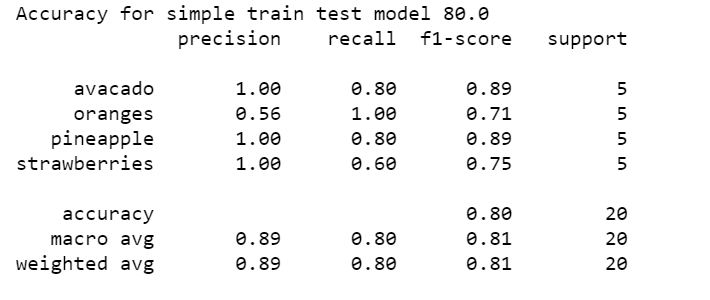
Description automatically generated

**2.4.3. Neural Network Classifiers**

Using different combinations of n\_components of PCA, below table describes accuracy results for simple train, test model and model using gridsearch with cross validation approach.

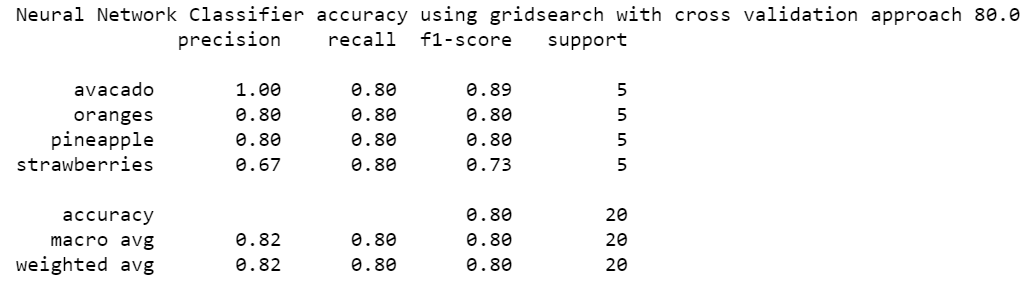
|  |  |  |
| --- | --- | --- |
| PCA(n\_components) | Simple Train/Test Model Accuracy | Model using GridSearch Accuracy |
| 5 | 75 | 85 |
| 10 | 85 | 85 |
| 20 | 80 | 70 |
| 40 | 55 | 50 |

The output we obtained using Neural Network Classifier using simple test/train model:

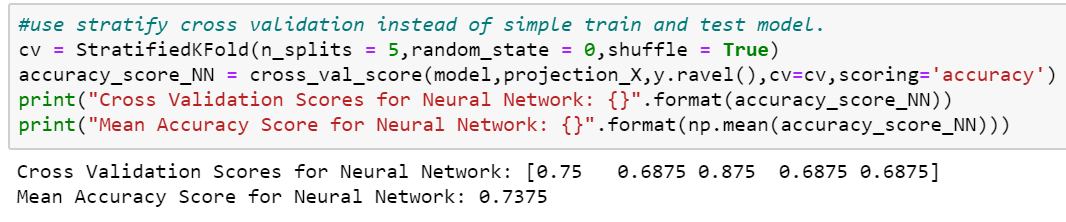


The output we obtained for Neural Network Classifier using GridSearch

with cross validation is



Stratify cross validation score



ROC curve for Neural Network classifier

A close up of a map

Description automatically generated

**2.5. Conclusion**

SVM (accuracy score 92%) and Random Forest classifier (accuracy score 92%) perform better than Neural Network classifier (accuracy score 80%) in the image classification task with sample size of 80. Based on ROC curve and AUC, Neural Network performs better than SVM and Random Forest classifier.

Using stratify cross validation, mean accuracy score for Neural Network(73.5%) is less than accuracy score using Simple train/test model (80%) and model created using GridSearch with cross validation approach (80%) .For SVM mean accuracy score (90%) is higher than accuracy score using Simple train/test model (87.5%) but less than model created using GridSearch with cross validation approach(92%). For Random Forest Classifier mean accuracy score (92.5%) is slightly higher than the accuracy score using Simple train/test model(92%) and model(92%) created using GridSearch with cross validation approach respectively. Hence using stratify cross validation only for SVM with simple train/test model accuracy score increased.

Using GridSearch with cross validation approach,for Neural Network classifier it doesn’t changes the accuracy. In the case of Random Forest classifier, accuracy remains the same but with SVM classifier accuracy is increased. Hence for SVM classifier gridsearch with cross validation approach yields higher value in accuracy.

Since dataset is small(sample size=80),both SVM and Random Forest Classifier performs better than Neural Network model in classify an image both using simple train/test model and model created using GridSearch with cross validation approach.

In the case of simple train/test model, Random Forest classifier performs better than SVM and Neural Network.

**2.6 References:**

Lecture Notes of the class

https://en.wikipedia.org/wiki/RMS\_Titanic

<https://www.history.com/topics/early-20th-century-us/titanic>

<https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html>

<https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/>

<https://pillow.readthedocs.io/en/3.1.x/reference/Image.html>

<https://scikit-learn.org/stable/>

<https://numpy.org/>