Ira A. Fulton Schools of Engineering
Arizona State University

CSE 572: Data Mining

Project 3: Supervised Machine Learning

Purpose

In this project you will use supervised learning techniques to classify the independent component (IC) images of resting state functional magnetic resonance imaging (rs-fMRI) scans as Noise and Resting state network (RSN).

Objectives

Students will be able to:

- Develop code to train a machine model on images for binary classification
- Assess accuracy of machine model

Technology Requirements

Python 3.6 to 3.9

Project Description

In this project, you will write a python program to train a machine model which is best for binary classification of provided IC images as Noise or RSN. A training dataset of 5 patients is provided. Every patient data is accompanied by a label list. Label list contains label 0,1,2,3. Label '0' refers to Noise IC and **anything greater than 0 (i.e., label 1, 2 and 3) refers to RSN.**

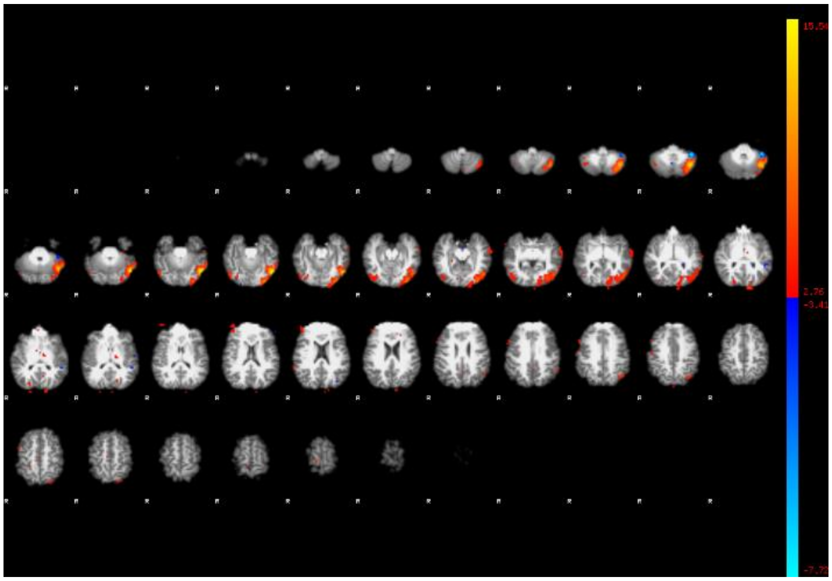


Figure 1: Independent Component (IC)

Directions

There are two main parts to the process:

1. Creating a single label type of RSN IC images. RSN ICs have different labels (1, 2 and 3). You must first create a list where only two labels would be present: ‘0’ for Noise and ‘1’ for RSN. For this change any labels greater than 0 as 1.
2. Once you have the correct label lists ready for the provided five patients dataset, you can apply any supervised learning technique of your choice which will perform best on binary classification of the Noise and RSN IC images.

Dataset:

You will be given five patient’s dataset which will contain approximately 100 spatial ICs of that patient’s rs-fMRI scan. All the scans will look like figure 1 (except the blood activation (red/blue clusters) part).

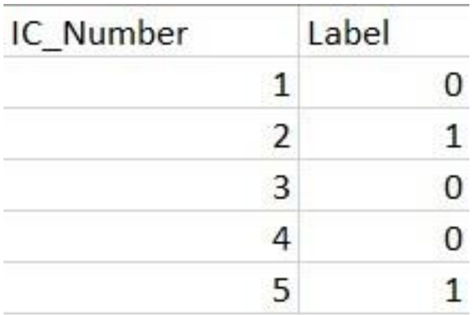
Submission Directions for Project Deliverables

Submit your Python code in a zip file on the Canvas. Please submit a zipped file containing your code files as

“**yourfirstname_lastname_Assignment1.zip**". Do not create an additional folder; just zip the files directly. Submit two python files,

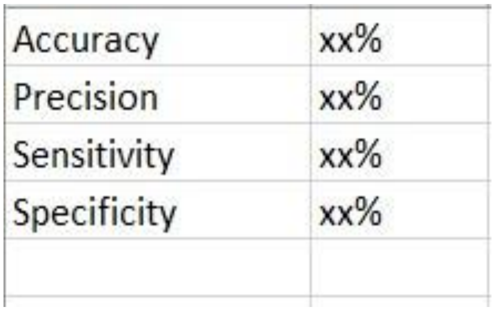
a) classification.py
b) test.py
c) Trained model

The classification.py will read all the images (images those end with word “thresh”) and labels from the given data, change the labels 1,2 and 3 as ‘1’ for the RSN images and train & generate a machine learning model based on the labels to classify the IC image as RSN or Noise. RSN IC image refers to label ‘1’ and Noise IC image refers to label ‘0’. Load trained model to test.py and make test.py read the ‘testPatient’ folder as we did in the previous assignments. The ‘testPatient’ folder will further have ‘test_Data’ folder which will have IC images similar to the provided data and the ‘testPatient’ folder will also have a ‘test_Labels’ csv file which will have the labels of test patient data (in the similar format of the provided label data). Your ‘test.py’ will output two csv files: “**Results.csv**” which will contain the labels classified as 0 (Noise) or 1 (RSN) for every IC image of the” testPatient” (see figure 2 for reference) and “**Metrics.csv**” which will contain the results of metrics in the percentage format (see figure 3 for reference). For example, if the “testPatient” data contains 120 IC images, then your code should output two csv files, first “**Results.csv**” with 120 rows which will provide the labels for every IC image as 0 (Noise) or 1 (RSN) and second “**Metrics.csv**” which will provide metrics results of the test patient data.



IC_Number	Label
1	0
2	1
3	0
4	0
5	1

Figure 2



Accuracy	xx%
Precision	xx%
Sensitivity	xx%
Specificity	xx%

Figure 3

Evaluation

We will run your test.py on a new patient’s data (‘testPatient’). Your code should output two csv files “**Results.csv**” and “**Metrics.csv**”. Successful execution of your code will result in 70 points. The remaining 30 points will be based on the total number of metrics you got within the 10% error limit.