Term Project #2 Fall, 2021

Due Date: 11:59PM 01/14/2022

Data mining in PJI research: periprosthetic joint infection early diagnosis for total joint replacement

Introduction

In contrast to Project #1, which is about Data Preparation for Data Mining algorithms (i.e. C4.5), Project #2 is focused on Predictive Modeling. Your job is to conduct a comparative study of different predictive modeling algorithms, including (a) CNN (Convolution NN), (b) LG (Logistic Regression), (c) SVM (Support Vector Machine), (d) RF (Random Forest), and (e) C4.5 (you already did in Proj #1). As they may perform quite differently based on different model topologies (e.g. CNN) or hyperparameters (e.g. SVM), you are required to tune the models as much as you can. In the meanwhile, you are also required to prepare the "best" training data for these modeling algorithms to produce their "best" classifiers for PJI diagnosis.

Hints. To design a CNN, you may need to consider the number of convolution layers, the specification of a kernel (e.g. 3*3, 4*4, etc.), pooling methods, etc. For a SVM, what kernel functions to use (linear, polynomial, radial-based, etc.)? For RF, any limit to the number of decision trees?
There are many issues of modeling. It is your job to produce your best CNN, LG, SVM, RF and C4.5.

Data

You are provided with the same dataset as in Project #1 for training and validating your models. As mentioned above, prepare your "best" training data for each of the mining (learning) algorithms in the comparative study. Since these algorithms are based on different design philosophies, the "best" training data may not be all the same. You are expected to employ what you have learned from Project #1 and to prepare the "best" training data for each of the mining algorithms.

How to build the models

There is no need to do the modeling from scratch (e.g. no need to code a CNN from the very beginning). That is, you are allowed to construct your predictive models by using some packages, e.g., WEKA, scikit-learn, PyTorch, TensorFlow, etc. However, you are still required to work on the topologies, the hyperparameters, and all other potential factors that may affect the performances of your final models.
Note that there is a wide variety of ANN topologies (e.g. CNN, RNN, conventional MLP, LSTM, transformers, etc.), but for Project #2, you can only use CNN to demonstrate whether CNN can actually self-learn any useful feature in the convolution layers from the PJI data.

Evaluation of models: Performance Measures

Your models will be evaluated on a test data set (posted later on e3) based on the performance measures defined below.

Performance Metric	**Definition**[*]
Recall[a]	TP / (TP + FN)
Precision[b]	TP / (TP + FP)
Percentage Accuracy (ACC)	(TP + TN) / (TP + TN + FP + FN)
F1-score	$\frac{2 \times Recall \times Precision}{Recall + Precision}$
Matthews Correlation Coefficient (MCC)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}}$
Area Under Curve (AUC)	Area under the ROC curve

[*] TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.
[a] Recall is equivalent to sensitivity in its definition.
[b] Precision is equivalent to positive predictive value in its definition.

How your Project #2 to be graded

- Your final models (CNN, LG, SVM, RF and C4.5) will be tested on an external test data set (posted later on e3) for all the performance measures listed above.
- What to turn in
  - A report that describes the same and the different parts in the “best” data for the modeling algorithms.
  - A report that describes the topology or the hyperparameters of your models, e.g., # convolution layers, kernel size, pooling strategy, etc. for CNN; type of kernel, cost and gamma used in SVM, and any other important specification of your models.
  - For each model, turn in your final training dataset and test dataset in Excel sheet format.
  - Turn in a snapshot (screen shot) of partial prediction output from each model for the first test example in the test data set.
  - Convert the prediction output from each model for the first test example into human comprehensible if-then-else classification rules. If you think it is practically impossible, just say so in your report.