

CVanalysis Documentation

Description

This document contains descriptions of all objects that will be used for the cvAnalysis

Data Import and Processing

Code Document: *functionsDataImportProcessing.R*

phq9DataSubsetImport

- File Dependencies:
 - PHQ9subset.csv
 - * subsetted PHQ9 data exported from Excel
- File Exports:
 - phq9Subset.Rdata
 - * Rdata file containing final formatted variables
 - datin.Rdata
 - * Redundant Rdata file for experimentation and nomenclature convinience
- Goals:
 - Save important variables from full PHQ9 data for analysis
 - Calculate and integrate variables necessary for downstream analysis
 - Adds variables that will be employed for supervised learning
- Processes:
 - Reduces data to Question Responses with values 0-3
 - Calculates sum of responses for each subject
 - Adds a string identifier for each question

list(dat.train.out, dat.test.out, N.obs) = CVsplit(dat.in, n.set)

- Description: a function for partitioning the full PHQ9 data into a user-specified number of training and test sets.
- Arguments:
 - dat.in: data desired to partition (will generally always be the full PHQ9 data)
 - n.set: Number of sets (total, including test set) into which to partition data
 - * **NOTE:** setting n.set=2496 will return the entire data set
- Return Values:
 - Type is list()
 - format: `obj <- list(dat.train.out, dat.test.out, N.obs)`
 - dat.train.out: a list of training data sets (in order of selection)
 - dat.test.out: a list of testing data sets (in order of selection)
 - N.obs: a numerical value indicating the number of observations in a training/test set

Weight Calculations

Code Document: *functionsWeightCalculations.R*

c(C1, C2, C3) = Peval(dat.in, qNum, respNum, qName)

- Description: a function that calculates the probability of being classified into one of three different outcome categories based upon training data.
- Arguments:
 - dat.in: training data file on which to perform calculations
 - qNum: question number of inquiry
 - respNum: given response of inquiry
 - qName: Column String name for response of interest Q1, Q2, Q3...
- Return Values:
 - Type is vector c()
 - format: obj <- c(C1, C2, C3)
 - a three-item vector of probabilities corresponding to the chances of eventually being characterized within a certain classification outcome C1, C2, C3

list(Peval(resp0), Peval(resp1), Peval(resp2), Peval(resp3)) = PCVeval(dat.in, N.PCV.obs)

- Description: a function that evaluates Peval for all possible answers to a single question.
- Arguments:
 - dat.in: training data file on which to perform calculations
 - qNum: question number of inquiry
 - qName: Column String name for response of interest Q1, Q2, Q3...
- Return Values:
 - Type is list()
 - Format: obj <- list(Peval(resp0), Peval(resp1), Peval(resp2), Peval(resp3))
 - a four-item list of probabilities corresponding to the chances of eventually being characterized within a certain classification outcome C1, C2, C3 provided that you answered a user specified question with any one of the four responses

list(PCVeval(Q1),...,PCVeval(Q9)) = PCVeval_overQnum(dat.in, N.PCV.obs)

- Description: a function that evaluates PCVeval for questions in a data set specified.
- Arguments:
 - dat.in: training data file on which to perform calculations
- Return Values:
 - Type is list()
 - Format: obj <- list(PCVeval(Q1),..., PCVeval(Q9))
 - a nine-item list, each of which contains the four-item lists from the evaluation of the PCVeval() function.

out.list = ReformatWeights(in.list)

- Description: This script defines a function that will take the output created by the PCVeval_overQnum() function and transform the output list into a single data frame for easier computation.
- Arguments:
 - list.in: list argument 9X4X3 dimensional array produced by PCVeval_overQnum()
- Returns:
 - out.list: List of 3 data frames, one list with three 9X4 data frames

Probabilistic Score Calculations

Code Document: *functionsProbScoreCalc.R*

EvalSeqData(cvData.train.in, cvData.test.in, cvData.N.set.in)

- Descriptions: Defines a function that will generalize the evaluation of the probabilistic scoring algorithm across multiple subjects to multiple train/test data pairings
- Arguments:
 - cvData.train.in: List of training data sets as output by CVsplit
 - cvData.test.in: List of training data sets as output by CVsplit
 - cvData.N.set.in: the total number of partitions that the original data was split into.
- Returns:
 - out.seq.subj.probs.overSet: A list of cvData.N.set.in lists of subject-specific 3X9 sequential probability evaluations

Eval.Subjects.Prob.sequence = EvalSeqSubject(train.weights.in, test.data.in, set.no.in)

- Description: Defines a function that will generalize the evaluation of the probabilistic scoring algorithm to multiple subjects within a single test/train data pairing
- Arguments:
 - train.weights.in: Output of the ReformatWeights() function
 - test.data.in: corresponding data frame of test data used to calculate training weights
 - set.no.in: user-specified test-train pairing index (will generally be looping index)
- Returns:
 - Eval.Subjects.Prob.sequence: list of nine 1X3 probability calculations corresponding to the probability of being in C1, C2, C3, for each subject.

sub.Prob.sequence=EvalSeq(train.weights.in, test.data.in, set.no.in, subject.no.in)

- Description: Defines a function that will calculate the probabilistic scoring algorithm for a single subject in a single train/test data pairing
- Arguments:
 - train.weights.in: Output of the ReformatWeights() function

- test.data.in: corresponding data frame of test data used to calculate training weights
- set.no.in: user-specified test-train pairing index (will generally be looping index)
- subject.no.in: subject (in terms of row number in the test set) for which the probabilistic scoring algorithm is desired to be evaluated.
- Returns:
 - sub.Prob.sequence: list of 1X3 probability calculations corresponding to the probability of being in C1, C2, C3

Probabilistic Score Evaluations

Code Document: *functionsScoringAnalysis.R*

out.result<-list(index.j,max.seq.val,index.max)=convg(sub.seq.in, thresh.in)

- Description: Defines a function that determines if an subject specific input sequence of probabilities (3x9 probabilistic) converges to a class (C1, C2, C3) with a provided probability threshold.
- Arguments:
 - sub.seq.in: 3x9 sequence of probabilistic scores specific to an individual.
 - thresh.in: probabilistic convergence threshold (e.g. 90%, 95%, 99.5%)
- Returns:
 - index.j: how many iterations of the probabilistic scoring algorithm were required for the sequence to converge. (note that $\text{thresh.in} > \frac{1}{3} \Rightarrow \text{index.j} \in \{2, \dots, 9\}$)
 - max.seq.val: Probability at convergence
 - index.max: Class index at maximum probability (which class eventually converged to)