Yiduo Feng
CS 699
project
Date: 6/21/2022

Analysis of 2018 BRFSS Survey Data

CS 699 Final Project (Prof. Jae Young Lee, Summer 2022)

Dataset Details

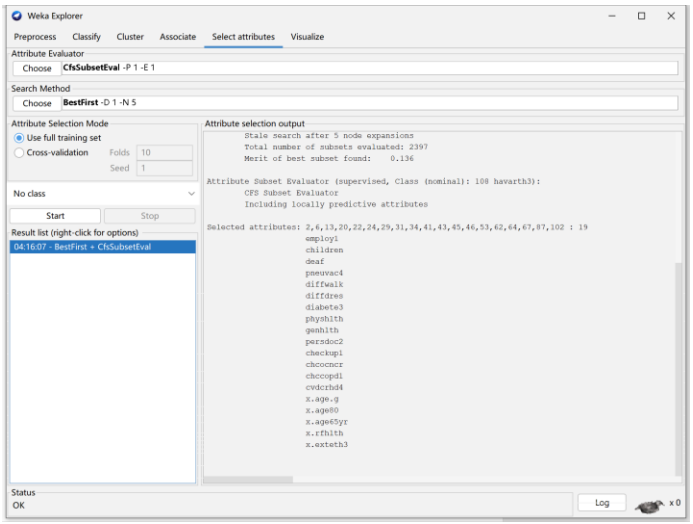
This data set is collected from the CDC website in 2018 on the diseases and physical conditions of people who participated in the survey. Each row in this dataset represents information about one participant, including 11933 tuples and 108 attributes.

Objective

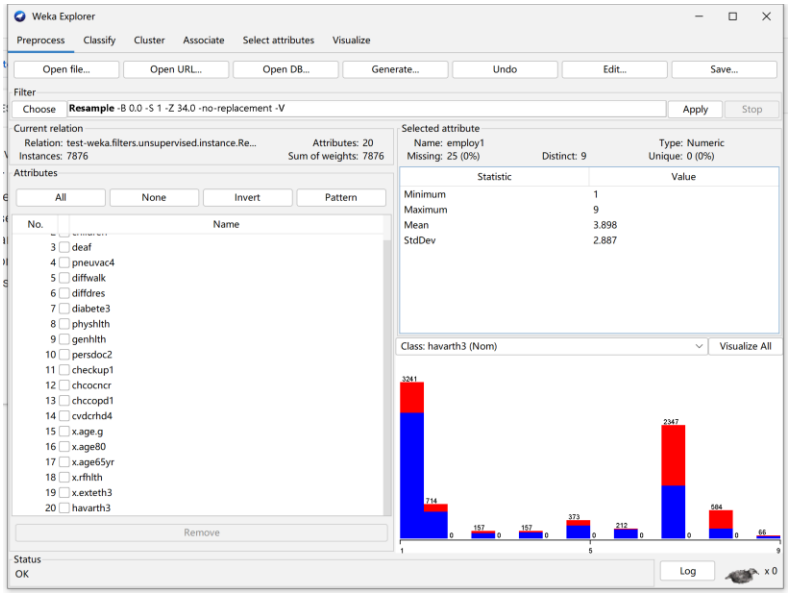
Divide the dataset into two parts: training dataset and test dataset, use five different attribute selection methods to reduce the dataset, and take out some attributes for classification. Each attribute selection method will perform five different classifications and will eventually test the total 25 classifier models, and we will find the best one with the best prediction.

Attribute selection methods 1:

Apply an attribute selection method on the *project-training.arff* dataset to select a subset of attributes. Save it as *reduced training dataset-1*. The selected attributes are shown in the picture below.
Attribute Evaluator: **CfsSubsetEval**
The value of a subset of attributes is evaluated by the predictive efficiency of each feature as well as its redundancy.
Search Method : **BestFirst**
Model the dataset through a greedy algorithm, select the n best features, and select the features that yield the model with the best performance.
By using this attribute selection method, we got 19 attributes.
They are employ1, children, deaf, pneuvac4, diffwalk, diffdres, diabete3, physhlth, genhlth, persdoc2, checkup1, chcocncr, chccopd1, cvdcrhd4, x.age.g, x.age80, x.age65yr, x.rfhlth, x.exteth3.



From the project-test. arff dataset, select only those attributes that are in the reduced training dataset, and save it as reduced test dataset-1.



(1) NaiveBayes

Based on Bayes' theorem, use Bayesian formula to calculate the relationship between attributes, features and classes for classification.

```
=== Summary ===

Correctly Classified Instances        2937               72.3934 %
Incorrectly Classified Instances      1120               27.6066 %
Kappa statistic                          0.4143
Mean absolute error                      0.2839
Root mean squared error                  0.4771
Relative absolute error                 63.4834 %
Root relative squared error            100.9023 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.735     0.299     0.829     0.735     0.779      0.420     0.794     0.879     2
               0.701     0.265     0.575     0.701     0.632      0.420     0.794     0.638     1
Weighted Avg.  0.724     0.287     0.743     0.724     0.729      0.420     0.794     0.798

=== Confusion Matrix ===

    a    b   <-- classified as
 1977  711 |    a = 2
  409  960 |    b = 1
```

(2) logistic regression

Like many other machine learning algorithms, logistic regression is borrowed from statistics. Logistic regression is the preferred method for binary classification tasks. It outputs a discrete binary result between 0 and 1. Base on there are four actions in this data set, and Logistic regression only have 2 kind of results, we consider action "Allow" is 0, and others is 1.

```
=== Summary ===

Correctly Classified Instances        3053               75.2526 %
Incorrectly Classified Instances      1004               24.7474 %
Kappa statistic                          0.4054
Mean absolute error                      0.3303
Root mean squared error                  0.4086
Relative absolute error                 73.8614 %
Root relative squared error             86.4235 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.885    0.508    0.774      0.885    0.826      0.416    0.804     0.882     2
                 0.492    0.115    0.686      0.492    0.573      0.416    0.804     0.654     1
Weighted Avg.    0.753    0.375    0.744      0.753    0.741      0.416    0.804     0.805

=== Confusion Matrix ===

    a    b   <-- classified as
 2379  309 |    a = 2
  695  674 |    b = 1
```

(3) SGD

The full name of sgd algorithm is stochastic gradient descent, which has the advantage of faster convergence than batch gradient descent. The word "random" in the name of the algorithm is the central spirit of the algorithm.

```
=== Summary ===

Correctly Classified Instances        3017               74.3653 %
Incorrectly Classified Instances      1040               25.6347 %
Kappa statistic                          0.3633
Mean absolute error                      0.2563
Root mean squared error                  0.5063
Relative absolute error                 57.3258 %
Root relative squared error            107.0786 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.910    0.583    0.754      0.910    0.825      0.386    0.664     0.746     2
                 0.417    0.090    0.702      0.417    0.523      0.386    0.664     0.490     1
Weighted Avg.    0.744    0.417    0.737      0.744    0.723      0.386    0.664     0.659

=== Confusion Matrix ===

    a    b   <-- classified as
 2446  242 |    a = 2
  798  571 |    b = 1
```

(4) SimpleLogistic

Similar to Logistic, SimpleLogistic uses LogitBoost, while Logistic uses a Ridge estimator.

```
=== Summary ===

Correctly Classified Instances        3042               74.9815 %
Incorrectly Classified Instances      1015               25.0185 %
Kappa statistic                          0.4004
Mean absolute error                      0.3283
Root mean squared error                  0.4077
Relative absolute error                 73.4068 %
Root relative squared error             86.2234 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.881    0.507    0.773      0.881    0.823      0.410    0.807     0.887     2
                 0.493    0.119    0.678      0.493    0.571      0.410    0.807     0.664     1
Weighted Avg.    0.750    0.376    0.741      0.750    0.738      0.410    0.807     0.812

=== Confusion Matrix ===

    a    b   <-- classified as
 2367  321 |    a = 2
  694  675 |    b = 1
```

(5) VotedPerceptron

The method is based on the intuition that since a prediction vector produces more correct predictions, it should have a larger weight.

```
=== Summary ===

Correctly Classified Instances        2859               70.4708 %
Incorrectly Classified Instances      1198               29.5292 %
Kappa statistic                          0.2914
Mean absolute error                      0.2953
Root mean squared error                  0.5434
Relative absolute error                 66.0338 %
Root relative squared error            114.923  %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.848    0.576    0.743      0.848    0.792      0.299    0.636     0.731     2
               0.424    0.152    0.586      0.424    0.492      0.299    0.642     0.453     1
Weighted Avg.  0.705    0.433    0.690      0.705    0.691      0.299    0.638     0.637

=== Confusion Matrix ===

   a    b   <-- classified as
 2279  409 |    a = 2
  789  580 |    b = 1
```

Attribute selection methods 2:

Attribute Evaluator: **GainRatioAttributeEval**

By calculating the gain ratio relative to the class, the value of the attribute is evaluated.

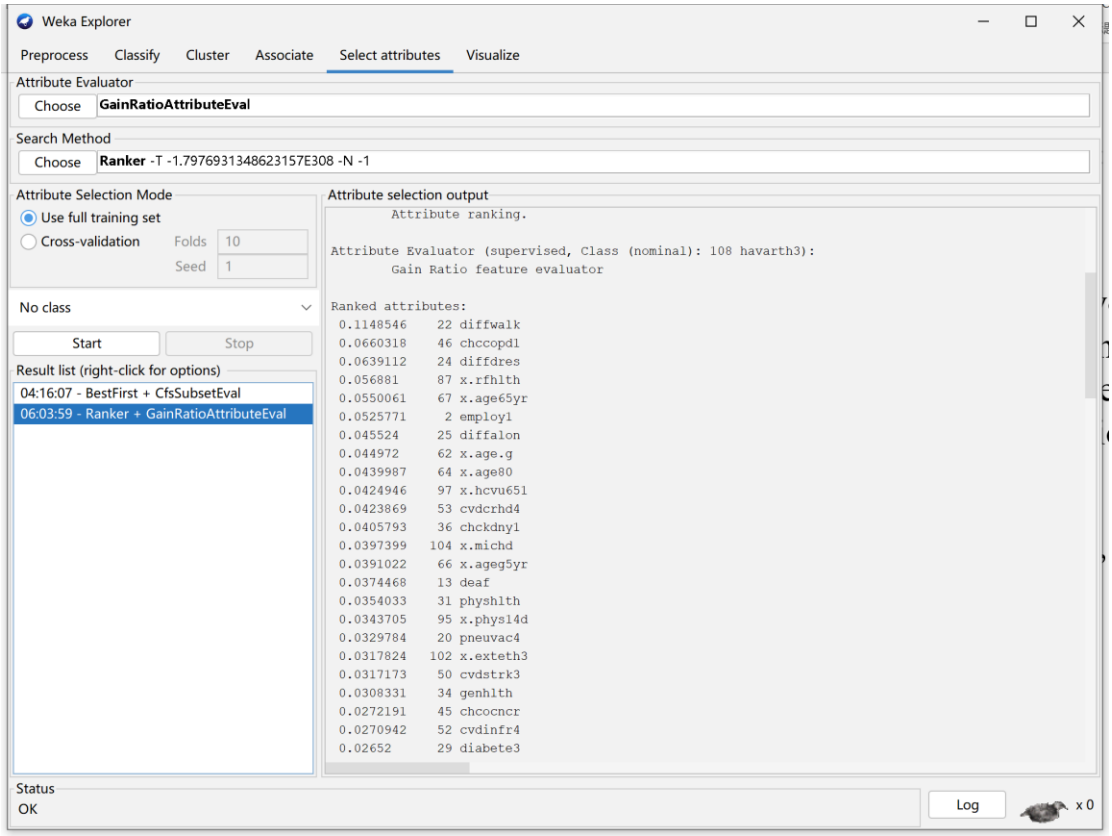
Search Method : **Ranker**

Rank attributes by their assessed value.

Because this method provides all properties and it ranks properties by value. Here I choose the top ten attributes to reduce the dataset.

The attributes are:

```
0.1148546    22 diffwalk
0.0660318    46 chccopd1
0.0639112    24 diffdres
0.056881     87 x.rfhlth
0.0550061    67 x.age65yr
0.0525771     2 employ1
0.045524     25 diffalon
0.044972     62 x.age.g
0.0439987    64 x.age80
0.0424946    97 x.hcvu651
```



(1) IBk

Given a training data set, for a new input instance, find the K instances closest to the instance in the training data set, and most of these K instances belong to a certain class, classify the input instance into this class.

```
=== Summary ===

Correctly Classified Instances        2914               71.8265 %
Incorrectly Classified Instances      1143               28.1735 %
Kappa statistic                          0.3064
Mean absolute error                      0.3318
Root mean squared error                  0.4622
Relative absolute error                 74.2069 %
Root relative squared error             97.758  %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.882    0.604    0.741      0.882    0.806      0.322    0.723     0.805     2
               0.396    0.118    0.632      0.396    0.487      0.322    0.723     0.560     1
Weighted Avg.  0.718    0.440    0.704      0.718    0.698      0.322    0.723     0.722

=== Confusion Matrix ===

    a    b   <-- classified as
 2372  316 |    a = 2
  827  542 |    b = 1
```

(2) AdaBoostM1

According to the classification error rate of the current classifier, the distribution of sample weights is adjusted to ensure that the weight of misclassified samples increases and the weight of correctly classified samples decreases; and the weight of the current classifier in the final decision is adjusted.

```
=== Summary ===

Correctly Classified Instances        2993               73.7737 %
Incorrectly Classified Instances      1064               26.2263 %
Kappa statistic                          0.3361
Mean absolute error                      0.3443
Root mean squared error                  0.4188
Relative absolute error                 76.985  %
Root relative squared error             88.571  %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.923    0.626    0.743      0.923    0.823      0.368    0.775     0.847     2
                 0.374    0.077    0.712      0.374    0.490      0.368    0.775     0.603     1
Weighted Avg.    0.738    0.441    0.733      0.738    0.711      0.368    0.775     0.765

=== Confusion Matrix ===

    a    b   <-- classified as
 2481  207 |    a = 2
  857  512 |    b = 1
```

(3) Bagging

Techniques for reducing generalization error by combining several models. The main idea is to train several different models separately, and then let all models vote on the output of the test examples.

```
=== Summary ===

Correctly Classified Instances        2981               73.4779 %
Incorrectly Classified Instances      1076               26.5221 %
Kappa statistic                          0.348
Mean absolute error                      0.3402
Root mean squared error                  0.4152
Relative absolute error                 76.0873 %
Root relative squared error             87.8188 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.894    0.577    0.753      0.894    0.817      0.365    0.789     0.878     2
                 0.423    0.106    0.669      0.423    0.518      0.365    0.789     0.639     1
Weighted Avg.    0.735    0.418    0.724      0.735    0.716      0.365    0.789     0.797

=== Confusion Matrix ===

    a    b   <-- classified as
 2402  286 |    a = 2
  790  579 |    b = 1
```

(4) FilteredClassifier

A class that runs a classifier on data that has passed the filter.

```
=== Summary ===

Correctly Classified Instances        2984               73.5519 %
Incorrectly Classified Instances      1073               26.4481 %
Kappa statistic                          0.3395
Mean absolute error                      0.3615
Root mean squared error                  0.4253
Relative absolute error                 80.8464 %
Root relative squared error             89.9364 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.909    0.605    0.747      0.909    0.820      0.364    0.736     0.799     2
                 0.395    0.091    0.688      0.395    0.502      0.364    0.736     0.573     1
Weighted Avg.    0.736    0.431    0.727      0.736    0.713      0.364    0.736     0.723

=== Confusion Matrix ===

    a    b   <-- classified as
 2443  245 |    a = 2
  828  541 |    b = 1
```

(5) LogitBoost

The method of integrating several classifiers into one classifier, before the boosting algorithm was generated, there were two more important methods of

integrating multiple classifiers into one classifier.

```
=== Summary ===

Correctly Classified Instances        2992               73.7491 %
Incorrectly Classified Instances      1065               26.2509 %
Kappa statistic                          0.3481
Mean absolute error                      0.3402
Root mean squared error                  0.4152
Relative absolute error                 76.0831 %
Root relative squared error             87.8154 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.905    0.592    0.750      0.905    0.820      0.370    0.787     0.865     2
               0.408    0.095    0.687      0.408    0.512      0.370    0.787     0.630     1
Weighted Avg.  0.737    0.424    0.729      0.737    0.716      0.370    0.787     0.786

=== Confusion Matrix ===

    a    b   <-- classified as
 2433  255 |    a = 2
  810  559 |    b = 1
```

Attribute selection methods 3:

Attribute Evaluator: **InfoGainAttributeEval**

The value of the attribute is obtained by calculating the information gain.

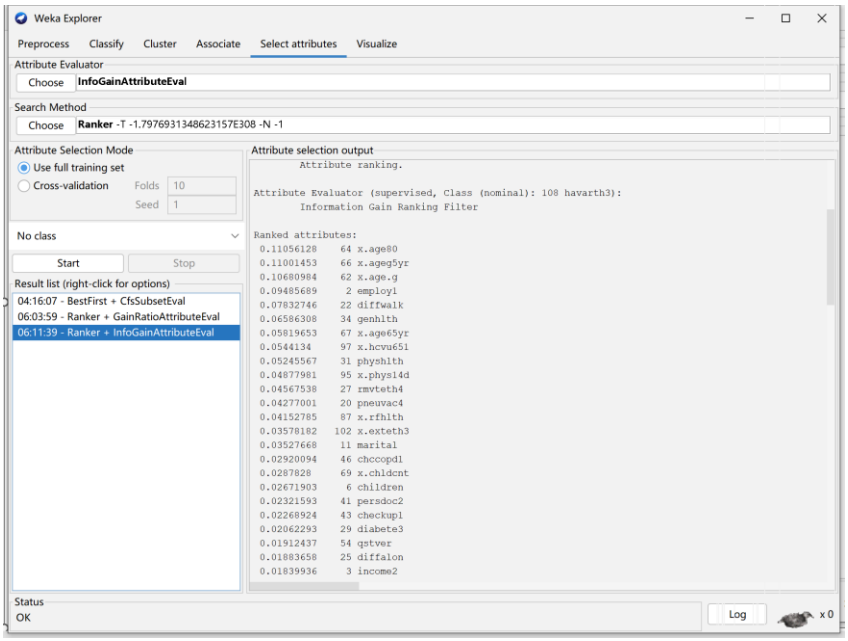
Search Method : **Ranker**

Rank attributes by their assessed value.

Because this method provides all properties and it ranks properties by value. Here I choose the top ten attributes to reduce the dataset.

The attributes are:

```
0.11056128    64 x.age80
0.11001453    66 x.ageg5yr
0.10680984    62 x.age.g
0.09485689     2 employ1
0.07832746    22 diffwalk
0.06586308    34 genhlth
0.05819653    67 x.age65yr
0.0544134     97 x.hcvu651
0.05245567    31 physhlth
0.04877981    95 x.phys14d
```



(1) MultiClassClassifier

A meta-classifier for processing multi-class datasets with binary classifiers.

```
=== Summary ===

Correctly Classified Instances        3044               75.0308 %
Incorrectly Classified Instances      1013               24.9692 %
Kappa statistic                          0.4004
Mean absolute error                      0.3339
Root mean squared error                  0.4086
Relative absolute error                 74.6788 %
Root relative squared error             86.4107 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.883    0.510    0.773      0.883    0.824      0.411    0.803     0.885     2
                 0.490    0.117    0.681      0.490    0.570      0.411    0.803     0.659     1
Weighted Avg.    0.750    0.377    0.742      0.750    0.738      0.411    0.803     0.809

=== Confusion Matrix ===

    a    b   <-- classified as
 2373  315 |    a = 2
  698  671 |    b = 1
```

(2) MultiScheme

Select a classifier class from multiple classifiers to obtain cross-validation and performance on training data.

```
=== Summary ===

Correctly Classified Instances        2688               66.2559 %
Incorrectly Classified Instances      1369               33.7441 %
Kappa statistic                          0
Mean absolute error                      0.4472
Root mean squared error                  0.4728
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.663      1.000    0.797      ?        0.500     0.663     2
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.337     1
Weighted Avg.    0.663    0.663    ?          0.663    ?          ?        0.500     0.553

=== Confusion Matrix ===

    a    b   <-- classified as
 2688    0 |    a = 2
 1369    0 |    b = 1
```

(3) RandomCommittee

Build an ensemble of random classifiers.

```
=== Summary ===

Correctly Classified Instances        2861               70.5201 %
Incorrectly Classified Instances      1196               29.4799 %
Kappa statistic                          0.2879
Mean absolute error                      0.3286
Root mean squared error                  0.4673
Relative absolute error                 73.4869 %
Root relative squared error             98.8235 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.855    0.588    0.740      0.855    0.793      0.297    0.714     0.798     2
                 0.412    0.145    0.591      0.412    0.485      0.297    0.714     0.543     1
Weighted Avg.    0.705    0.439    0.690      0.705    0.689      0.297    0.714     0.712

=== Confusion Matrix ===

    a    b   <-- classified as
 2297  391 |    a = 2
  805  564 |    b = 1
```

(4) RandomizableFilteredClassifier

Build an ensemble of random classifiers.

```
=== Summary ===

Correctly Classified Instances        2840               70.0025 %
Incorrectly Classified Instances      1217               29.9975 %
Kappa statistic                          0.2797
Mean absolute error                      0.3268
Root mean squared error                  0.4994
Relative absolute error                 73.091  %
Root relative squared error            105.6241 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.845    0.584    0.739      0.845    0.789      0.287    0.698     0.793     2
                 0.416    0.155    0.577      0.416    0.483      0.287    0.698     0.527     1
Weighted Avg.    0.700    0.440    0.685      0.700    0.686      0.287    0.698     0.703

=== Confusion Matrix ===

    a    b   <-- classified as
 2271  417 |    a = 2
  800  569 |    b = 1
```

(5) RandomSubSpace

It is a type of ensemble learning. Random subspace reduces the correlation between each classifier by training each classifier with a random subset of features instead of all features.

```
=== Summary ===

Correctly Classified Instances        3037               74.8583 %
Incorrectly Classified Instances      1020               25.1417 %
Kappa statistic                          0.3891
Mean absolute error                      0.356
Root mean squared error                  0.4153
Relative absolute error                 79.6149 %
Root relative squared error             87.8369 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.893    0.535    0.766      0.893    0.825      0.404    0.793     0.878     2
                 0.465    0.107    0.689      0.465    0.555      0.404    0.793     0.647     1
Weighted Avg.    0.749    0.390    0.740      0.749    0.734      0.404    0.793     0.800

=== Confusion Matrix ===

    a    b   <-- classified as
 2400  288 |    a = 2
  732  637 |    b = 1
```

Attribute selection methods 4:

Attribute Evaluator: **CorrelationAttributeEval**

By calculating the correlation relative to the class, the value of the attribute is evaluated.

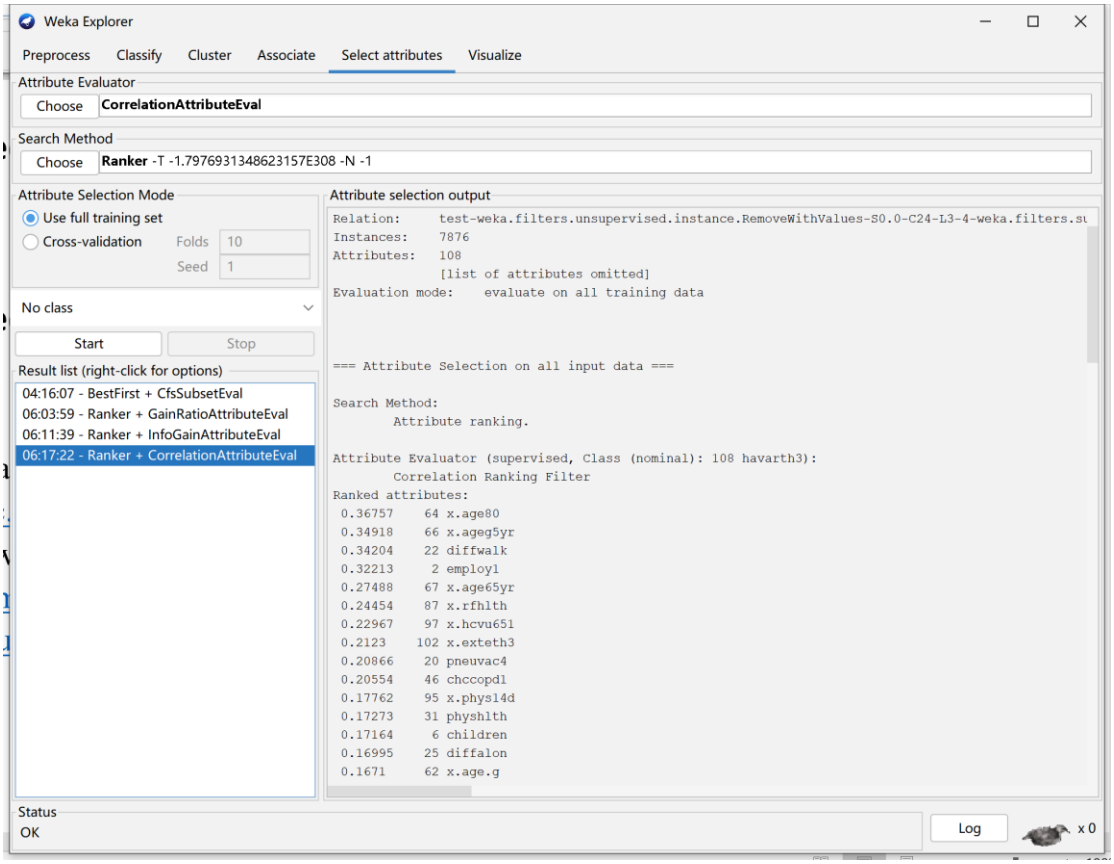
Search Method : **Ranker**

Rank attributes by their assessed value.

Because this method provides all properties and it ranks properties by value. Here I choose the top ten attributes to reduce the dataset.

The attributes are:

```
0.36757    64 x.age80
0.34918    66 x.ageg5yr
0.34204    22 diffwalk
0.32213     2 employ1
0.27488    67 x.age65yr
0.24454    87 x.rfhlth
0.22967    97 x.hcvu651
0.2123    102 x.exteth3
0.20866    20 pneuvac4
0.20554    46 chccopd1
```



(1) Stacking

Take the answer of the base model as input, and let the secondary learner learn to organize the assignment of weights to the answer of the base model.

```
=== Summary ===

Correctly Classified Instances        2688               66.2559 %
Incorrectly Classified Instances      1369               33.7441 %
Kappa statistic                          0
Mean absolute error                      0.4472
Root mean squared error                  0.4728
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.663      1.000    0.797      ?        0.500     0.663     2
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.337     1
Weighted Avg.    0.663    0.663    ?          0.663    ?          ?        0.500     0.553

=== Confusion Matrix ===

    a    b   <-- classified as
 2688    0 |    a = 2
 1369    0 |    b = 1
```

(2) Vote

If an element appears, if it is equal to the candidate element, the votes of the candidate element are increased by 1; if it is not equal to the candidate element, the votes of the candidate element are decreased by 1.

```
=== Summary ===

Correctly Classified Instances        2688               66.2559 %
Incorrectly Classified Instances      1369               33.7441 %
Kappa statistic                          0
Mean absolute error                      0.4472
Root mean squared error                  0.4728
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.663      1.000    0.797      ?        0.500     0.663     2
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.337     1
Weighted Avg.    0.663    0.663    ?          0.663    ?          ?        0.500     0.553

=== Confusion Matrix ===

    a    b   <-- classified as
 2688    0 |    a = 2
 1369    0 |    b = 1
```

(3) WeightedInstancesHandlerWrapper

A wrapper around a classifier to enable weighted instance support.

```
=== Summary ===

Correctly Classified Instances        2688               66.2559 %
Incorrectly Classified Instances      1369               33.7441 %
Kappa statistic                          0
Mean absolute error                      0.4472
Root mean squared error                  0.4728
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.663      1.000    0.797      ?        0.500     0.663     2
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.337     1
Weighted Avg.    0.663    0.663    ?          0.663    ?          ?        0.500     0.553

=== Confusion Matrix ===

    a    b   <-- classified as
 2688    0 |    a = 2
 1369    0 |    b = 1
```

(4) InputMappedClassifier

The incompatibility problem is resolved by establishing a mapping between the structures on which the classifier is built.

```
=== Summary ===

Correctly Classified Instances        2688               66.2559 %
Incorrectly Classified Instances      1369               33.7441 %
Kappa statistic                          0
Mean absolute error                      0.4472
Root mean squared error                  0.4728
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.663      1.000    0.797      ?        0.500     0.663     2
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.337     1
Weighted Avg.    0.663    0.663    ?          0.663    ?          ?        0.500     0.553

=== Confusion Matrix ===

    a    b   <-- classified as
 2688    0 |    a = 2
 1369    0 |    b = 1
```

(5) DecisionTable

The process of recursively selecting the most characteristic feature and dividing the training data according to this feature so that there is a best classification process for each sub-data set.

```
=== Summary ===

Correctly Classified Instances        3004               74.0449 %
Incorrectly Classified Instances      1053               25.9551 %
Kappa statistic                          0.3603
Mean absolute error                      0.3428
Root mean squared error                  0.4151
Relative absolute error                 76.6687 %
Root relative squared error             87.7853 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.900     0.573     0.755      0.900    0.821      0.379    0.784     0.862     2
               0.427     0.100     0.685      0.427    0.526      0.379    0.784     0.636     1
Weighted Avg.  0.740     0.414     0.732      0.740    0.722      0.379    0.784     0.786

=== Confusion Matrix ===

   a    b   <-- classified as
 2420  268 |    a = 2
  785  584 |    b = 1
```

Attribute selection methods 5:

Attribute Evaluator: **SymmetricalUncertAttributeEval**

By calculating the Symmetrical uncertainty relative to the class, the value of the attribute is evaluated.

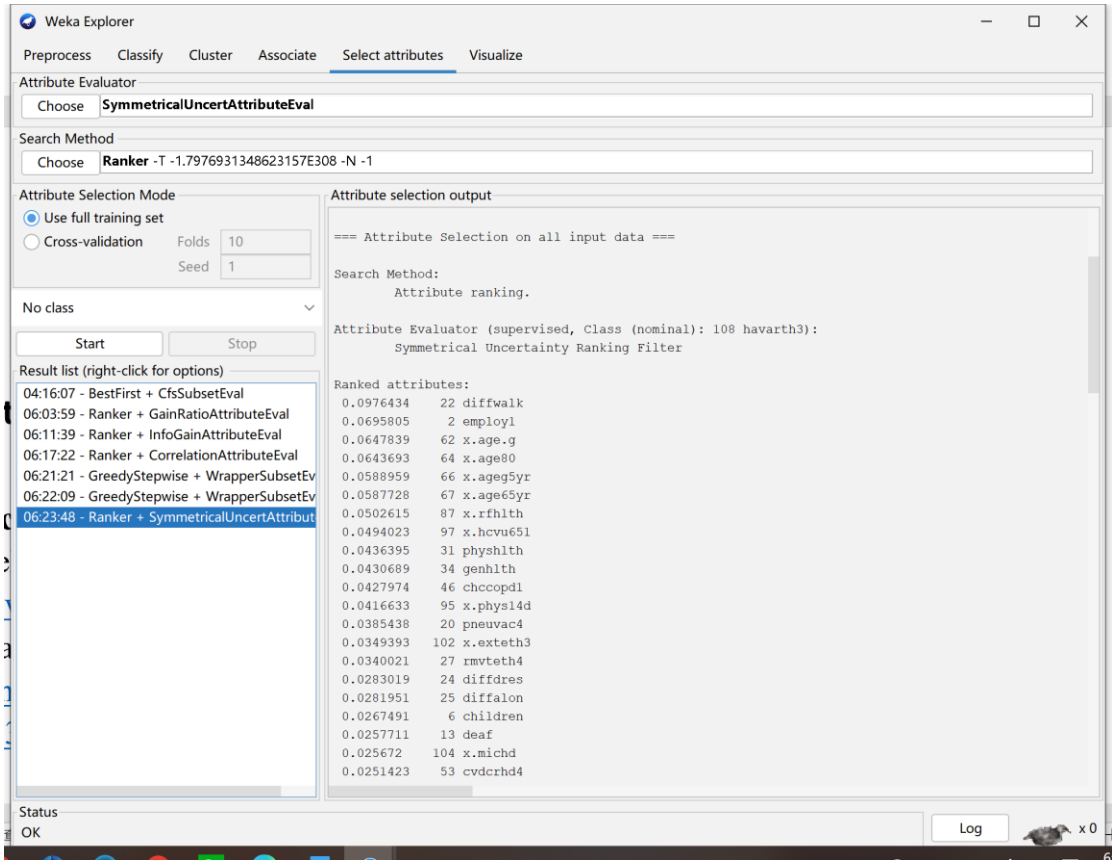
Search Method : **Ranker**

Rank attributes by their assessed value.

Because this method provides all properties and it ranks properties by value. Here I choose the top ten attributes to reduce the dataset.

The attributes are:

```
Ranked attributes:
 0.0976434    22 diffwalk
 0.0695805     2 employ1
 0.0647839    62 x.age.g
 0.0643693    64 x.age80
 0.0588959    66 x.ageg5yr
 0.0587728    67 x.age65yr
 0.0502615    87 x.rfhlth
 0.0494023    97 x.hcvu651
 0.0436395    31 physhlth
 0.0430689    34 genhlth
```



(1) JRip

Generate a rule that randomly divides non-covered instances into a growing set and a pruning set, specifying that each rule in the rule set has two rules to generate

```
=== Summary ===

Correctly Classified Instances        2974               73.3054 %
Incorrectly Classified Instances      1083               26.6946 %
Kappa statistic                          0.3585
Mean absolute error                      0.3767
Root mean squared error                  0.4359
Relative absolute error                 84.2331 %
Root relative squared error             92.1827 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.871    0.537    0.761      0.871    0.812      0.368    0.675     0.755     2
                 0.463    0.129    0.646      0.463    0.539      0.368    0.675     0.523     1
Weighted Avg.    0.733    0.399    0.722      0.733    0.720      0.368    0.675     0.677

=== Confusion Matrix ===

   a    b   <-- classified as
 2340  348 |    a = 2
  735  634 |    b = 1
```

(2) OneR

Find the feature with the best classification effect through the training set, use it as the classification basis, and use other features to calculate the error rate to realize the classification algorithm.

```
=== Summary ===

Correctly Classified Instances        2985               73.5765 %
Incorrectly Classified Instances      1072               26.4235 %
Kappa statistic                          0.321
Mean absolute error                      0.2642
Root mean squared error                  0.514
Relative absolute error                 59.0897 %
Root relative squared error            108.7135 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.935    0.656    0.737      0.935    0.824      0.361    0.640     0.732     2
                 0.344    0.065    0.730      0.344    0.468      0.361    0.640     0.473     1
Weighted Avg.    0.736    0.456    0.735      0.736    0.704      0.361    0.640     0.644

=== Confusion Matrix ===

    a    b   <-- classified as
 2514  174 |    a = 2
  898  471 |    b = 1
```

(3) PART

Create a rule to remove the instances covered by the rule, and then recursively create rules for the remaining instances until there are no remaining instances.

```
=== Summary ===

Correctly Classified Instances        2971               73.2315 %
Incorrectly Classified Instances      1086               26.7685 %
Kappa statistic                          0.3866
Mean absolute error                      0.3316
Root mean squared error                  0.4242
Relative absolute error                 74.1489 %
Root relative squared error             89.719  %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.823    0.446    0.784      0.823    0.803      0.388    0.772     0.847     2
                 0.554    0.177    0.615      0.554    0.583      0.388    0.772     0.600     1
Weighted Avg.    0.732    0.355    0.727      0.732    0.729      0.388    0.772     0.764

=== Confusion Matrix ===

    a    b   <-- classified as
 2212  476 |    a = 2
  610  759 |    b = 1
```

(4) J48

Based on top-to-bottom strategy, recursive divide-and-conquer strategy, select an attribute to place at the root node, generate a branch for each possible attribute value, divide the instance into multiple subsets, each subset corresponds to a branch of the root node , then recursively repeat the process on each branch. Stop when all instances have the same classification.

```
=== Summary ===

Correctly Classified Instances        3062               75.4745 %
Incorrectly Classified Instances       995               24.5255 %
Kappa statistic                          0.4065
Mean absolute error                      0.3377
Root mean squared error                  0.4185
Relative absolute error                 75.529  %
Root relative squared error             88.4985 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.894    0.518    0.772      0.894    0.828      0.420    0.777     0.844     2
                 0.482    0.106    0.698      0.482    0.570      0.420    0.777     0.604     1
Weighted Avg.    0.755    0.379    0.747      0.755    0.741      0.420    0.777     0.763

=== Confusion Matrix ===

    a    b   <-- classified as
 2402  286 |    a = 2
  709  660 |    b = 1
```

(5) RandomForest

Random forest is composed of many decision trees, and there is no relationship between different decision trees. When we carry out the classification task, new input samples enter, and each decision tree in the forest will be judged and classified separately. Each decision tree will get its own classification result, which one of the classification results of the decision tree is classified. At most, then the random forest will treat this result as the final result.

```
=== Summary ===

Correctly Classified Instances        2906               71.6293 %
Incorrectly Classified Instances      1151               28.3707 %
Kappa statistic                          0.3216
Mean absolute error                      0.3299
Root mean squared error                  0.442
Relative absolute error                 73.7837 %
Root relative squared error             93.4879 %
Total Number of Instances             4057

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.853    0.552    0.752      0.853    0.799      0.329    0.749     0.842     2
                 0.448    0.147    0.608      0.448    0.516      0.329    0.749     0.592     1
Weighted Avg.    0.716    0.415    0.703      0.716    0.704      0.329    0.749     0.757

=== Confusion Matrix ===

    a    b   <-- classified as
 2293  395 |    a = 2
  756  613 |    b = 1
```

Conclusion

According to all of the results of 25 classification models. Since in the five attribution selections, the number of 1s and the number of 2s in the class are quite different, we mentioned in the class discussion that ROC uses FP / (TN + FP), ROC will be affected by both FP and TN, so ROC is not accurate enough when the data differs greatly, so we ignore the value of ROC Area for this dataset, instead, we focus on PRC.

If we only look at the total accuracy, the top three are CfsSubsetEval method with logistic regression 75.26%, InfoGainAttributeEval with MultiClassClassifier 75.03% and SymmetricalUncertAttributeEval method with J48 75.47%. J48 has highest precision, recall, F-measure and MCC. Precision shows that J48 predicts the largest number of true positive samples TP/(TP+FP) among the positive samples, while Recall shows that TP/(TP+FN) positive samples are successfully predicted to be the largest number of positive samples. Based on the formula of F-measure, F-measure combines the results of precision and recall. The higher the F-measure, the more effective the test method is.

Thus, SymmetricalUncertAttributeEval attribute selection method with a J48 algorithm gave me the best performance.

References:

CDC. Behavioral Risk Factor Surveillance System

https://www.cdc.gov/brfss/annual_data/annual_2018.html
Blackboard. how-to-split-dataset-to-training-and-test.docx. https://onlinecampus.bu.edu/bbcswebdav/pid-10341678-dt-content-rid-70620893_1/courses/22sum1metcs699so1/course/syllabus/Supplements/stratified-split.pdf