CAMPUS
DE EXCELENCIA
INTERNACIONAL

Master Universitario en Ciencia de Datos

Practical Application 1

Classification

El Abbassi Widad

December 16th,2019

I. Introduction

In Today's world, Modernity has brought us ease regarding divorce. He is no longer as ashamed as he was several decades ago. It is a model that is both accepted and increasingly promoted and made tempting by advertising means. It's as if the model of the aging couple together has become a has been model that we no longer understand and most of divorce cases found today involves families that have children, which result in more impact on the well being of those children. Consequently, we find that this failing relationships do effect more family, friends and the surrounding more than it effect the couple itself. However, if divorce predictors of married couples can be estimated early, many divorces can be prevented.

II. Problem description

The goal of this article is the predictability of divorce among couples using the Divorce Predictors Scale (DPS) developed by Yöntem and İlhan (2017, 2018) on the basis of Gottman couples therapy (Gottman, 2014; Gottman and Gottman, 2012) According to this, the behaviors and reactions of couples in the minutes following an argument give an idea about divorce. Therefore, harsh start up in an argument is one of the most obvious predictors that a debate or marriage will not go well.

This project aims to evaluate and compare the accuracy of this Divorce Predictors Scale through several supervised and unsupervised classification algorithms through WEKA (Waikato Environment for Knowledge Analysis) software, which is widely used in data mining. For the supervised models, there will be four analyses: (1) with all original variables; (2) with a univariate filter feature subset selection; (3) with a multivariate (filter and wrapper) feature subset selection (4) using Metaclassifiers (stacking, bagging and boosting).
As for the **unsupervised** classification algorithms used (Simple Means, EM, cobweb, MakeDensityBasedCluster) , the evaluation of this clusters can be done using cluster to classes evaluation or ClassificationViaClustering method.

By solving this problem, we can get great insight into what the most significant features that can predict a non-promising relationships. These features could then be used by marriage counselors to increase the rate of divorces that can be prevented.

III. Methodology

1. Study Group

In this dataset, we have 170 samples with 54 input features. The data was gathered through a study

group of 170 participants where they had to fill in a “Personal Information Form” and “Divorce Predictors Scale” regarding gender, marital status, age, monthly income, family structure, type of marriage, happiness in marriage and divorce thought,

Among this participants, 84 (49%) were divorced and 86 (51%) were married couples where 84 (49%) of them were males and 86 females (51%). While the ages of the participants ranged from 20 to 63 ($\bar{X}$ = 36.04, SD = 9.34).

Furthermore, 74 (43.5%) were married for love, and 96 (56.5%) were married in an arranged marriage. While 127 (74.7%) of the participants had children, 43 (25.3%) had no children. In addition, it is also important to note that 18 (10.58%) of the participants were primary school graduate, 15 (8.8%) were secondary school graduate, 33 (19.41%) were high school graduate, 88 (51.76%) were college graduate, and 15 (8.8%) had master's degree.

The monthly incomes of the participants were as follows: 34 (20%) individuals had under 2000 TL, 54 (31.76%) had between 2001-3000 TL, 28 (16.47%) had between 3001-4000 TL and 54 (31.76%) individuals had a monthly income over 4000 TL

As a matter or Randomization, the data were collected from seven different regions of Turkey, predominantly from the Black Sea region (n=79).

2. Analysis of the Data

This dataset contains 54 features that on the basis of them we have to predict that the couple has been divorced or not:

1. If one of us apologizes when our discussion deteriorates, the discussion ends.
2. I know we can ignore our differences, even if things get hard sometimes.
3. When we need it, we can take our discussions with my spouse from the beginning and correct it.
4. When I discuss with my spouse, to contact him will eventually work.
5. The time I spent with my wife is special for us.
6. We don't have time at home as partners.
7. We are like two strangers who share the same environment at home rather than family.
8. I enjoy our holidays with my wife.
9. I enjoy traveling with my wife.
10. Most of our goals are common to my spouse.
11. I think that one day in the future, when I look back, I see that my spouse and I have been in harmony with each other.
12. My spouse and I have similar values in terms of personal freedom.
13. My spouse and I have similar sense of entertainment.
14. Most of our goals for people (children, friends, etc.) are the same.
15. Our dreams with my spouse are similar and harmonious.
16. We're compatible with my spouse about what love should be.
17. We share the same views about being happy in our life with my spouse
18. My spouse and I have similar ideas about how marriage should be
19. My spouse and I have similar ideas about how roles should be in marriage
20. My spouse and I have similar values in trust.
21. I know exactly what my wife likes.

22. I know how my spouse wants to be taken care of when she/he sick.
23. I know my spouse's favorite food.
24. I can tell you what kind of stress my spouse is facing in her/his life.
25. I have knowledge of my spouse's inner world.
26. I know my spouse's basic anxieties.
27. I know what my spouse's current sources of stress are.
28. I know my spouse's hopes and wishes.
29. I know my spouse very well.
30. I know my spouse's friends and their social relationships.
31. I feel aggressive when I argue with my spouse.
32. When discussing with my spouse, I usually use expressions such as ‘you always’ or ‘you never’ .
33. I can use negative statements about my spouse's personality during our discussions.
34. I can use offensive expressions during our discussions.
35. I can insult my spouse during our discussions.
36. I can be humiliating when we discussions.
37. My discussion with my spouse is not calm.
38. I hate my spouse's way of open a subject.
39. Our discussions often occur suddenly.
40. We're just starting a discussion before I know what's going on.
41. When I talk to my spouse about something, my calm suddenly breaks.
42. When I argue with my spouse, ı only go out and I don't say a word.
43. I mostly stay silent to calm the environment a little bit.
44. Sometimes I think it's good for me to leave home for a while.
45. I'd rather stay silent than discuss with my spouse.
46. Even if I'm right in the discussion, I stay silent to hurt my spouse.
47. When I discuss with my spouse, I stay silent because I am afraid of not being able to control my anger.
48. I feel right in our discussions.
49. I have nothing to do with what I've been accused of.
50. I'm not actually the one who's guilty about what I'm accused of.
51. I'm not the one who's wrong about problems at home.
52. I wouldn't hesitate to tell my spouse about her/his inadequacy.
53. When I discuss, I remind my spouse of her/his inadequacy.
54. I'm not afraid to tell my spouse about her/his incompetence

3. Preprocessing:

Like any other machine learning project , the first step “preprocessing” consists of detecting and dealing with missing values, encoding the categorical variables (in our case, 1 represent Divorced, value 0 represent not divorced).

4. Feature selection and classifiers choice:

In this step, we will choose several classification algorithms (probabilistic and non –probabilistic) that fit the best our problem, for example (ID3 cannot be used in this case seeing that it can only deal with nominal variables which not our case) and then compare their results to see whether one of them outperforms the other. The classifiers used in this project are presented as follow:

- **Supervised Classification:**

Non-probabilistic classifiers:

- **k-nearest neighbors**
- **Ripper Algorithm**

- **C4.5 Algorithm**
- **linear SVM**
- **Multilayer Perceptron**

Probabilistic classifiers:

- **Logstic Regression**
- **Naive Bayes**
- **Tree augmented Naive Bayes**

- **Feature Subset Selection**

- **Filter approach**

This method consists of using an attribute evaluator and a ranker to rank all the features in the dataset. However, it's interesting to note that the weights put by the ranker algorithms are different than those by the classification algorithm. Two steps are involved:

- Searching method
- Ranker

This Filter technique is best used in data mining where the goal is to reduce the number of feature from thousands down to hundred or fifty, seeing that with this method is it possible to experience an overfitting of the classifier we are designing or trying to develop.

- **Wrapper approach**

The wrapper method on the other hand consist of an evaluation method and a search method. The goal of this technique is to determine which feature performs the best with a particular algorithm.

In WEKA, we distinguish two wrapper methods:

- ClassifierSubsetEval : Use à classifier, specified in the object editor as a parameter, to evaluate sets of attributes on the training data or on a separate holdout set.

- WrapperSubsetEval : Also use a classifier to evaluate attribute sets, but employ cross-validation to estimate the accuracy of the learning scheme for each set.

- **Metaclassifiers**

Metaclassifiers have the advantage of combining several classification algorithms power to have as a result a more reliable model to classify and identify instances from, however, we must also note that multiple research paper have stated that combining multiple classifiers results in overfitting. Also instead of facing one classification algorithm problem, we will be facing the combination of several algorithms problems.

- **Ensembles (Bagging and Boosting)**
- **Voting**
- **Stacking**

- **Non-Supervised Classification:**

- **SimpleKMeans Clustering**
- **EM Clustering**
- **MakeDensityBasedClusterer**

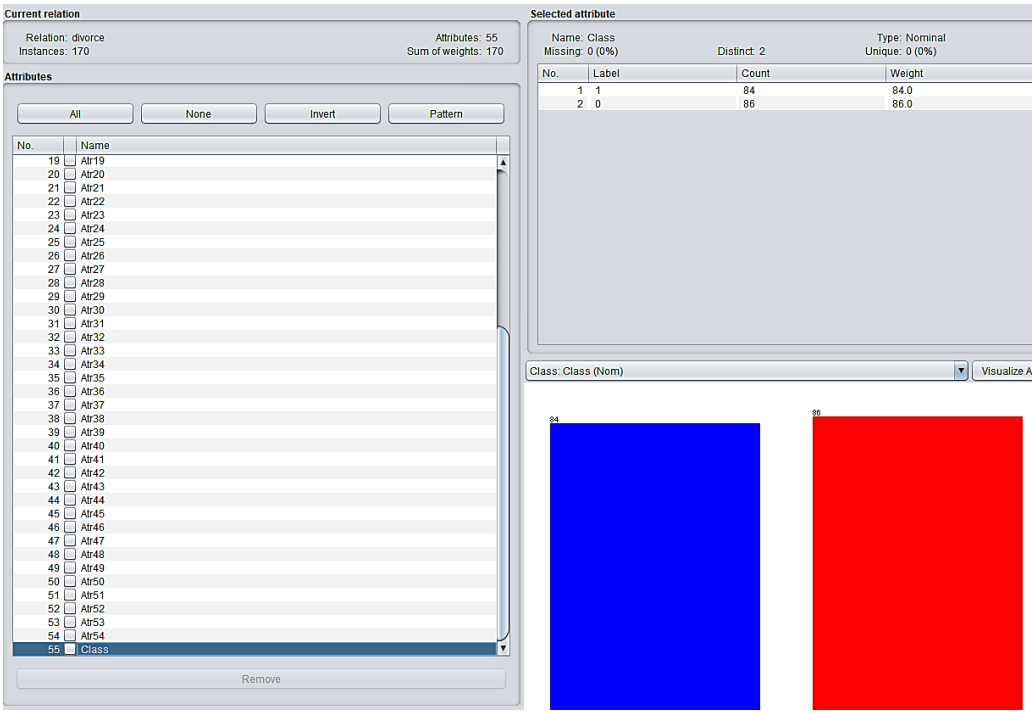
5. Comparaison & Evaluation Metrics:

- ✓ Accuracy: Accuracy is the ratio of number of correct predictions to the total number of input samples. So accuracy for each model has been displayed for each.
- ✓ Confusion Matrix : is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.
- ✓ Recall : literally is how many of the true positives were recalled.
- ✓ Precision : is how many of the returned hits were true positives.
- ✓ F-measure : combination of precision and recall.

In WEKA, The Experiment Environment allows us to perform statistical tests on the different performance measures to allow us to draw conclusions from the experiment. It will compare each algorithm in pairs and make some reasonable assumptions about the distribution of the results collected, such as that they are drawn from a Gaussian distribution. The significance level is set in the "Significance" parameter and is default to 0.05 (5%), which again, is just fine.

IV. Results

The classification methods were directly applied to this dataset and as a result we obtain as follow:



1. Supervised classifiers (Non-probabilistic)

Instead of splitting the dataset into Training and Testing sets, it is preferable to use cross validation with 10 folds which is the thing that wasn't used in several articles that were published in this field (The Hazards of Predicting Divorce Without Crossvalidation, Richard e. Heyman and amy m.Smith slep).

```
=== Summary ===

Correctly Classified Instances         166               97.6471 %
Incorrectly Classified Instances         4                2.3529 %
Kappa statistic                          0.9529
Mean absolute error                      0.0289
Root mean squared error                  0.1527
Relative absolute error                  5.7884 %
Root relative squared error             30.5261 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,000     1,000      0,952     0,976      0,954    0,972     0,977     1
               1,000     0,048     0,956      1,000     0,977      0,954    0,972     0,945     0
Weighted Avg.  0,976     0,024     0,978      0,976     0,976      0,954    0,972     0,961

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  0 86 |  b = 0
```

k-nearest neighbor

```
=== Summary ===

Correctly Classified Instances         162               95.2941 %
Incorrectly Classified Instances         8                4.7059 %
Kappa statistic                          0.9059
Mean absolute error                      0.0528
Root mean squared error                  0.2159
Relative absolute error                 10.5552 %
Root relative squared error             43.1616 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,047     0,952      0,952     0,952      0,906    0,947     0,936     1
               0,953     0,048     0,953      0,953     0,953      0,906    0,947     0,918     0
Weighted Avg.  0,953     0,047     0,953      0,953     0,953      0,906    0,947     0,927

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  4 82 |  b = 0
```

RIPPER

```
=== Summary ===

Correctly Classified Instances         162               95.2941 %
Incorrectly Classified Instances         8                4.7059 %
Kappa statistic                          0.9059
Mean absolute error                      0.0493
Root mean squared error                  0.2173
Relative absolute error                  9.8533 %
Root relative squared error             43.4552 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,964     0,058     0,942      0,964     0,953      0,906    0,950     0,926     1
               0,942     0,036     0,964      0,942     0,953      0,906    0,950     0,931     0
Weighted Avg.  0,953     0,047     0,953      0,953     0,953      0,906    0,950     0,928

=== Confusion Matrix ===

  a  b   <-- classified as
 81  3 |  a = 1
  5 81 |  b = 0
```

C4.5

```
=== Summary ===

Correctly Classified Instances         166               97.6471 %
Incorrectly Classified Instances         4                2.3529 %
Kappa statistic                          0.9529
Mean absolute error                      0.0235
Root mean squared error                  0.1534
Relative absolute error                  4.7048 %
Root relative squared error             30.6689 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,000     1,000      0,952     0,976      0,954    0,976     0,976     1
               1,000     0,048     0,956      1,000     0,977      0,954    0,976     0,956     0
Weighted Avg.  0,976     0,024     0,978      0,976     0,976      0,954    0,976     0,966

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  0 86 |  b = 0
```

Linear SVM

Multilayer perceptron:

Weka has a graphical interface that lets you create your own network structure with as many perceptron and connections as you like. And in this case, it uses backpropagation to learn a multi-layer perceptron to classify instances.

```
Correctly Classified Instances         166               97.6471 %
Incorrectly Classified Instances         4                2.3529 %
Kappa statistic                          0.9529
Mean absolute error                      0.0226
Root mean squared error                  0.1375
Relative absolute error                  4.529  %
Root relative squared error             27.4866 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,000     1,000      0,952     0,976      0,954    0,996     0,997     1
               1,000     0,048     0,956      1,000     0,977      0,954    0,996     0,996     0
Weighted Avg.  0,976     0,024     0,978      0,976     0,976      0,954    0,996     0,996

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  0 86 |  b = 0
```

Avec :

batch size =100 ;

Learning rate=0.3 ;

Hiddenlayer = (attribs + classes) /2 ;

2. Supervised classifiers (probabilistic)

```
Correctly Classified Instances         166               97.6471 %
Incorrectly Classified Instances         4                2.3529 %
Kappa statistic                          0.9529
Mean absolute error                      0.0234
Root mean squared error                  0.1528
Relative absolute error                  4.6856 %
Root relative squared error             30.5422 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,000     1,000      0,952     0,976      0,954    0,993     0,994     1
               1,000     0,048     0,956      1,000     0,977      0,954    0,993     0,993     0
Weighted Avg.  0,976     0,024     0,978      0,976     0,976      0,954    0,993     0,993

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  0 86 |  b = 0
```

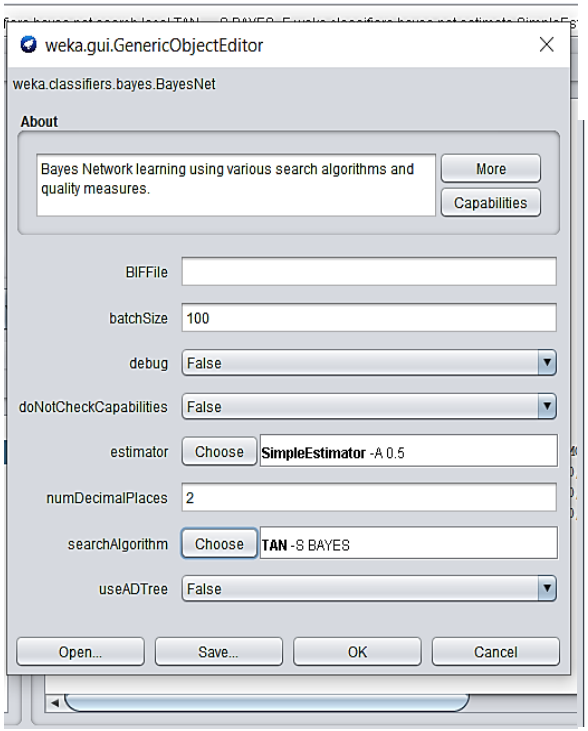
Logstic Regression

```
=== Summary ===

Correctly Classified Instances         166               97.6471 %
Incorrectly Classified Instances         4                2.3529 %
Kappa statistic                          0.9529
Mean absolute error                      0.0235
Root mean squared error                  0.1534
Relative absolute error                  4.7048 %
Root relative squared error             30.6689 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Cla
               0,952     0,000     1,000      0,952     0,976      0,954    0,999     0,999     1
               1,000     0,048     0,956      1,000     0,977      0,954    0,982     0,966     0
Weighted Avg.  0,976     0,024     0,978      0,976     0,976      0,954    0,991     0,982

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  0 86 |  b = 0
```

Naive Bayes

TAN (Tree augmented naive bayes):

For this algorithm, we must specify the TAN inside the configuration parameters of Bayes Network as shown bellow :




```
=== Summary ===

Correctly Classified Instances          56               96.5517 %
Incorrectly Classified Instances         2                3.4483 %
Kappa statistic                          0.931
Mean absolute error                      0.0337
Root mean squared error                  0.1816
Relative absolute error                  6.7412 %
Root relative squared error             36.3063 %
Total Number of Instances               58

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  C
               0,931     0,000     1,000      0,931     0,964      0,933    0,995     0,996     1
               1,000     0,069     0,935      1,000     0,967      0,933    0,995     0,995     0
Weighted Avg.  0,966     0,034     0,968      0,966     0,965      0,933    0,995     0,995

=== Confusion Matrix ===

  a  b   <-- classified as
 27  2 |  a = 1
  0 29 |  b = 0
```

TAN Output

3. Feature selection

3.1 Wrapper method

We use a subset evaluator that will create all possible subsets from the feature vector, and then it will select a classification algorithm to induce classifiers from the features in each subset.

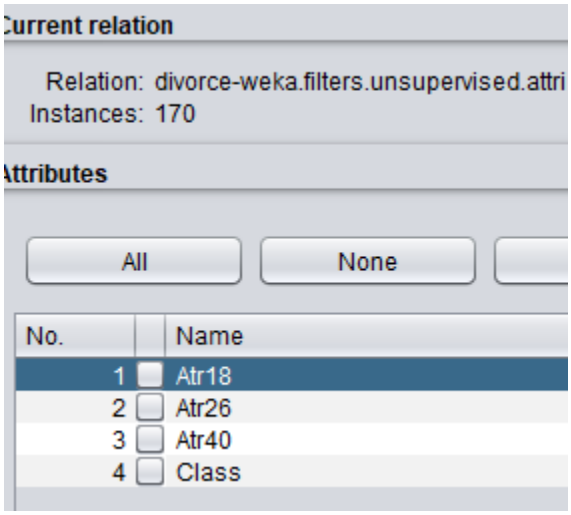
Here, we used the whole training dataset, so that the algorithm won't select different features each time if it was applied in cross validation.

Attribute Evaluator: ClassifierSubsetEval

Search method: BestFirst / Forward search

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 410
        Merit of best subset found:    1

Attribute Subset Evaluator (supervised, Class (nominal): 55 Class):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.bayes.NaiveBayes
        Scheme options:
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 18,26,40 : 3
                     Atr18
                     Atr26
                     Atr40
```

With backward search multiple features were displayed, the reason why I didn't continue working with it. After this step, in the preprocess tab we select only the features 18,26,40 that were obtained and apply the NaiveBayes Classifier in the classification tab after, which is the same that was chosen during this process in the ClassifierSubsetEval tab.




```
Correctly Classified Instances         170               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0035
Root mean squared error                  0.0328
Relative absolute error                  0.6935 %
Root relative squared error              6.5553 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000
Weighted Avg.  1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

  a  b   <-- classified as
 84  0 |  a = 1
  0 86 |  b = 0
```

And as we can see, we've got this time 100% precision, recall for both classes, which prove that excluding the other features gives a better results.

3.2Filter method :

For the filter technique, we going to use the InfoGainAttributeEval method ,we must use Ranker search method instead of best first, because in this case we are not searching for any subset, our goal is to evaluate every feature in our feature vector.

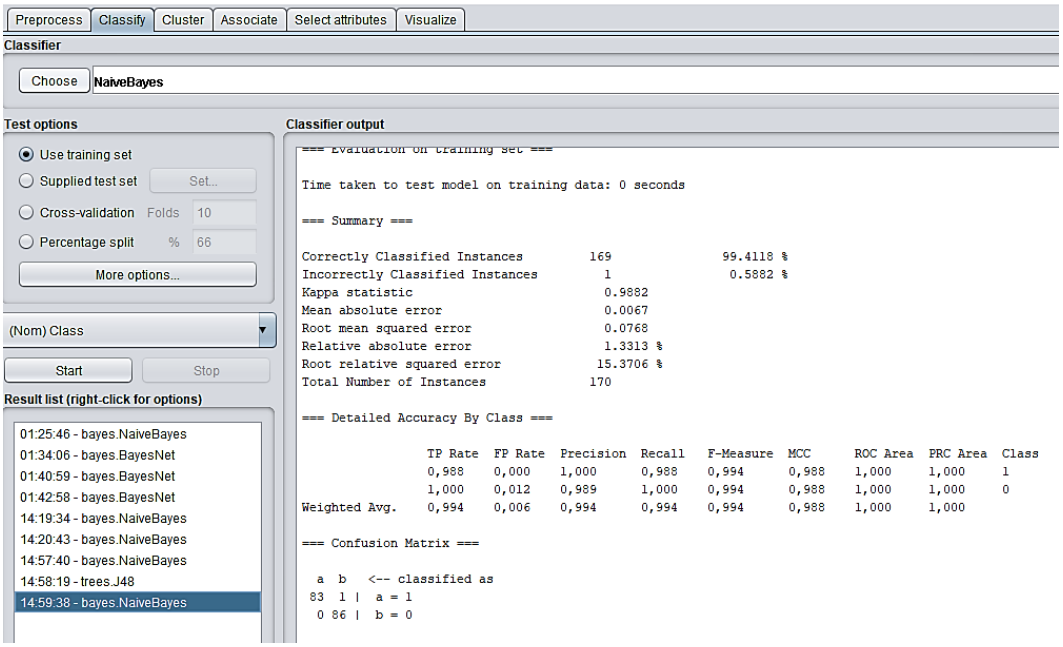
The ranker in the second step is responsible for ranking attributes by their individual evaluations. Use in conjunction with attribute evaluators (ReliefF, GainRatio, Entropy etc).

Since we have a large number of feature in this dataset, I decided to reduce the number of ranked feature to 5 instead of displaying them all

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 55 Class):
        Information Gain Ranking Filter

Ranked attributes:
 0.897    20 Atr20
 0.889    18 Atr18
 0.884    40 Atr40
 0.861    11 Atr11
 0.843    19 Atr19

Selected attributes: 20,18,40,11,19 : 5
```

Filter method result

Again, going back to the preprocess tab, excluding the last two features and classifying with Naïve Bayes, we obtain:



Results with taking account only selected feature

4. Metaclassifier :

4.1 ensembles:

Inside AdaBoostM1or bagging, we have the methods of more than one classification algorithms.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         166               97.6471 %
Incorrectly Classified Instances         4                2.3529 %
Kappa statistic                          0.9529
Mean absolute error                      0.0557
Root mean squared error                  0.1495
Relative absolute error                 11.1426 %
Root relative squared error             29.8935 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,000     1,000       0,952   0,976      0,954    0,997     0,998     1
               1,000     0,048     0,956       1,000   0,977      0,954    0,997     0,997     0
Weighted Avg.  0,976     0,024     0,978       0,976   0,976      0,954    0,997     0,997

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  0 86 |  b = 0
```

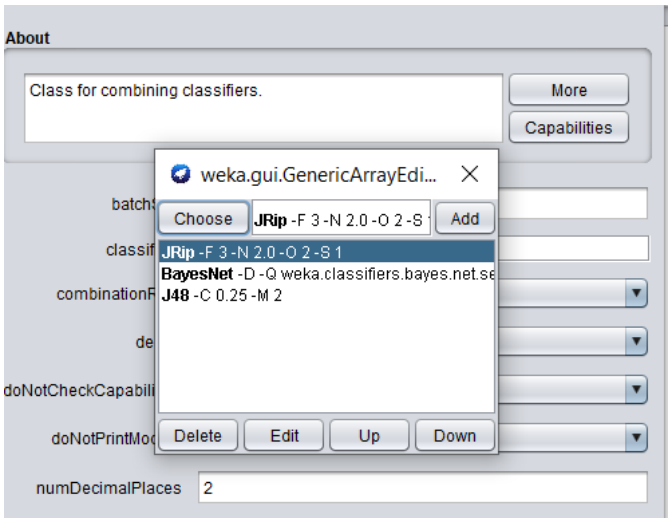
Bagging

```
=== Summary ===

Correctly Classified Instances         164               96.4706 %
Incorrectly Classified Instances         6                3.5294 %
Kappa statistic                          0.9294
Mean absolute error                      0.0357
Root mean squared error                  0.1857
Relative absolute error                  7.1411 %
Root relative squared error             37.1296 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,952     0,023     0,976       0,952   0,964      0,930    0,988     0,991     1
               0,977     0,048     0,955       0,977   0,966      0,930    0,988     0,986     0
Weighted Avg.  0,965     0,036     0,965       0,965   0,965      0,930    0,988     0,988

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  2 84 |  b = 0
```

AdaBoostM1

4.2 Voting :

Voting method whereby base-learners are made different by training them over slightly different training sets. First, we need to modify the algorithm from one to several ones.

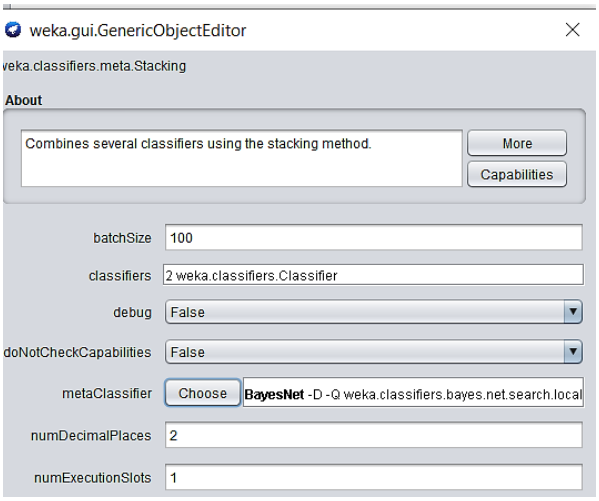



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         164               96.4706 %
Incorrectly Classified Instances         6                3.5294 %
Kappa statistic                          0.9294
Mean absolute error                      0.0419
Root mean squared error                  0.1695
Relative absolute error                  8.3711 %
Root relative squared error             33.8915 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  C
               0,964     0,035     0,964       0,964   0,964      0,929    0,973     0,983     1
               0,965     0,036     0,965       0,965   0,965      0,929    0,970     0,937     0
Weighted Avg.  0,965     0,035     0,965       0,965   0,965      0,929    0,971     0,960

=== Confusion Matrix ===

  a  b   <-- classified as
 81  3 |  a = 1
  3 83 |  b = 0
```

Voting Output

4.3 Stacking :

Select Naïve bayes in the configuration parameter of meta => stacking.




```
Correctly Classified Instances         168               98.8235 %
Incorrectly Classified Instances         2                1.1765 %
Kappa statistic                          0.9765
Mean absolute error                      0.0122
Root mean squared error                  0.1084
Relative absolute error                  2.4476 %
Root relative squared error             21.6767 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,976    0,000    1,000      0,976    0,988      0,977    1,000     1,000     1
               1,000    0,024    0,977      1,000    0,989      0,977    1,000     1,000     0
Weighted Avg.  0,988    0,012    0,989      0,988    0,988      0,977    1,000     1,000

=== Confusion Matrix ===

  a  b   <-- classified as
 82  2 |  a = 1
  0 86 |  b = 0
```

Stacking output

The results differ depending on the algorithms chosen, for another manipulation (combining Bayesian Network, JRip,MultilayerPerceptron)

The results obtained are shown below:

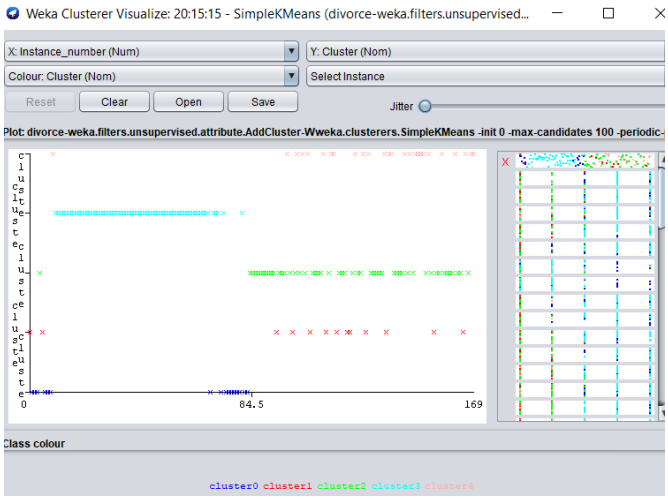
```
=== Summary ===

Correctly Classified Instances         161               94.7059 %
Incorrectly Classified Instances         9                5.2941 %
Kappa statistic                          0.8941
Mean absolute error                      0.0637
Root mean squared error                  0.2284
Relative absolute error                 12.7395 %
Root relative squared error             45.6737 %
Total Number of Instances              170

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
               0,952    0,058    0,941      0,952    0,947      0,894    0,946     0,929
               0,942    0,048    0,953      0,942    0,947      0,894    0,946     0,918
Weighted Avg.  0,947    0,053    0,947      0,947    0,947      0,894    0,946     0,924

=== Confusion Matrix ===

  a  b   <-- classified as
 80  4 |  a = 1
  5 81 |  b = 0
```

5. Non-Supervised Classification

5.1 SimpleKMeans Clustering

In SimpleKmeans, we define the number of cluster we would like to have, in this case I will begin with 5, and then try the results for 3.

But the problem here, is that one of the attribute will be the class itself, and it doesn't seem fair to include the class while doing the clustering, therefor we need to exclude the class attribute.

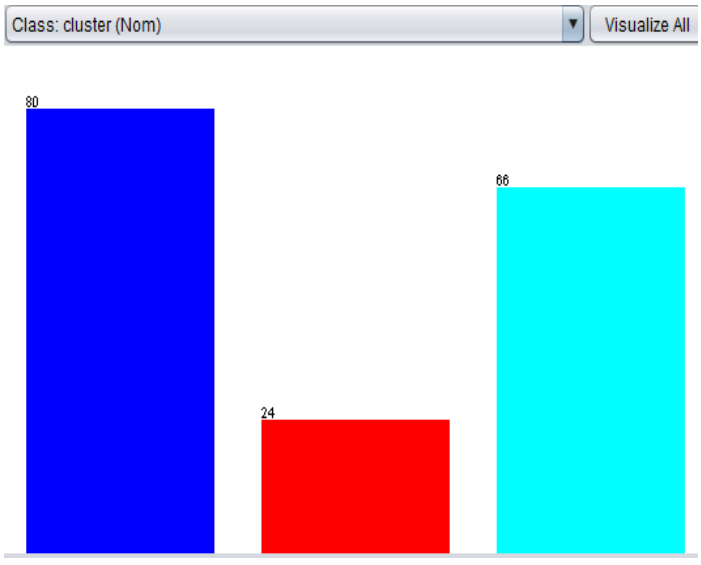


Cluster =5

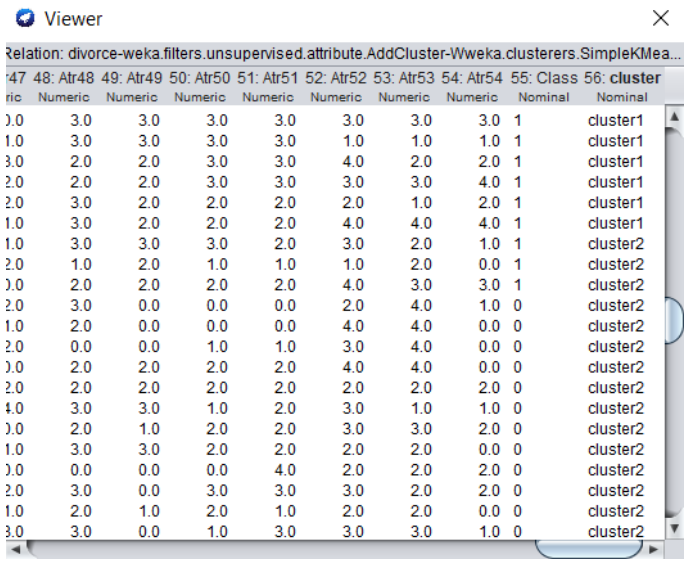
cluster=3

Now that we could visualize the cluster created, what if we would like to know which instances does a cluster contain?

In order to visualize that, we need to apply AddCluster as filter on the dataset at the preprocess tab with selecting the same number of cluster we did in the cluster tab (cluster=3), this functionality will add a new feature called cluster as shown below, and then we can compare each cluster to the assigned class in the Edit button.



number of cluster



classes and their correspondant cluster

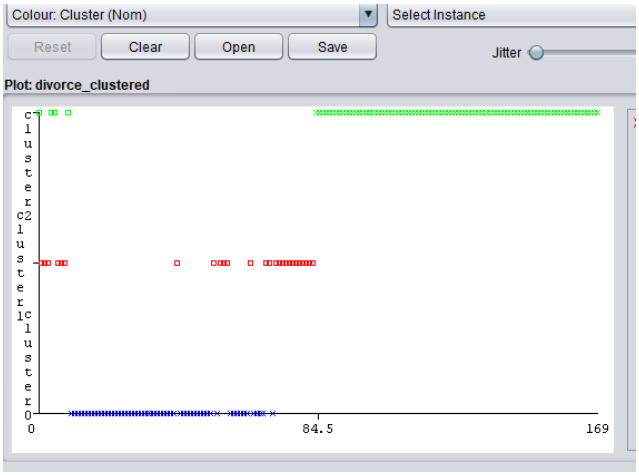
Going back to cluster tab, we see the results obtained for simpleKmeans :

```
=== Model and evaluation on training set ===

Clustered Instances

0       80 ( 47%)
1       24 ( 14%)
2       66 ( 39%)


Class attribute: Class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 80  3  1 | 1
  0 21 65 | 0

Cluster 0 <-- 1
Cluster 1 <-- No class
Cluster 2 <-- 0

Incorrectly clustered instances :       25.0     14.7059 %
```

5.2 EM (Expectation Maximization):

The goal of this clustering algorithm is to maximize the overall probability or likelihood of the data, given the final clusters instead of maximizing the differences in means for continuous variables.

```
Class attribute: Class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 54 26  4 | 1
  0  0 86 | 0

Cluster 0 <-- 1
Cluster 1 <-- No class
Cluster 2 <-- 0

Incorrectly clustered instances :       30.0     17.6471 %
```



EM output

5.3 MakeDensityBasedClusterer

Here the identification of clusters is made by a grouping of point density, a cluster is made by region of high point of density and each cluster is separated on others by the low point density.

```
Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       80 ( 47%)
1       25 ( 15%)
2       65 ( 38%)


Log likelihood: -56.29058
```

Density Based clustering output

Also it can be translated as number of incorrectly clustered instances: 81.0 - 47.6471 %

5.4 HierarchicalClusterer

Again, with HierarchicalClusterer we fixed the number of clusters to = 3 and obtained the below results which are obviously worse than the previous ones(all the instances were classified as one).

```
=== Clustering model (full training set) ===

Cluster 0
((((1.0:1.87083,((((((((0.0:1.65831,((((((((((((((((((((((((((((((((((((((((((((((((((1



Time taken to build model (full training data) : 0.15 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      168 ( 99%)
1        1 (  1%)
2        1 (  1%)
```

HierarchicalClusterer Output

6. Comparison:

6.1 Weka Experiment Environment

As matter of number of classifiers tested in this project, I choosed to compare the performances of only some supervised classifiers using Weka Experiment Environment. Comparison here is done by taking the first choosed algorithm against the other models.

For the configuration parameters:

Significance: 0.05% which means any performances difference found among the n classifiers, models will be within 5% confidence interval.

Performance measure: percent_correct – F-measure

Evaluation:

* : means significantly poor(Loss).

v: means significantly better (Victor).

```
Dataset                   (1) bayes.Ba | (2) funct (3) funct (4) meta. (5) meta. (6) meta. (7) meta.
----------------------------------------------------------------------------------------------------
divorce                 (100)   97.65 |   97.71     97.65     97.18     97.47     84.06 *   50.59 *
----------------------------------------------------------------------------------------------------
                                (v/ /*) |   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/0/1)   (0/0/1)


Key:
(1) bayes.BayesNet '-D -Q bayes.net.search.local.K2 -- -P 1 -S BAYES -E bayes.net.estimate.SimpleEstimator -- -A 0.5' 746037443258775954
(2) functions.LibSVM '-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"C:\\\\Program Files\\\\Weka-3-8\" -seed 1' 14172
(3) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210779
(4) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -1178107808933117974
(5) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0' -115879962237199703
(6) meta.ClassificationViaClustering '-W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A \"weka.core.Euclic
(7) meta.Stacking '-X 10 -M \"rules.ZeroR \" -S 1 -num-slots 1 -B \"trees.J48 -C 0.25 -M 2\" -B \"bayes.NaiveBayes \" -B \"bayes.BayesNet -D -Q bayes.net.search.local.K2 -- -P 1 -S BAYES
```

Comparison output

This results indicates that the first four classifiers are not significantly different (0/1/0) comparing them the first one which is the Bayes Network, the scores are more or less the same. The last two metaclassifier have an * character next to their results in the table, indicating that the results significantly poor (0/0/1). If an algorithm had results larger than the base algorithm and the difference was significant, a little “v” would appear next to the results.

6.2 Comparing Classsifiers Results

	Correctly classified instances	**Incorrectly classified instances**	**Kappa Statisctics**	**F Measure**	**Precision**	**Recall**
Supervised Classification – Non Probabilistic						
k-nearest neighbors	97.6471	2.3529	0.9529	0.976	0.978	0.976
Ripper Algorithm	95.2941	4.7059	0.9059	0.953	0.953	0.953
C4.5 Algorithm	95.4706	4.7059	0.9059	0,953	0,953	0,953
linear SVM	97.6471	2.3529	0.9529	0.976	0.976	0.976
Multilayer Perceptron	97.6471	2.3529	0.9529	0.978	0.976	0.976
Supervised Classification – Probabilistic						
Logstic Regression	97.6471	2.3529	0.9529	0,976	0,976	0,976
Naive Bayes	97.6471	2.3529	0.9529	0.978	0.976	0.976
Tree augmented Naive Bayes	96.5517	3.4483	0.931	0,965	0,968	0,966
MetClassifiers						
Bagging	97.6471	2.3529	0.9529	0.978	0.976	0.976
Boosting	96.4706	3.594	0.9294	0.965	0.965	0.965
Voting	96.4706	3.5294	0.9294	0,965	0,965	0,965
Stacking	98.8235	1.1765	0.9765	0,988	0,989	0,988
Non-Supervised Classification						
SimpleKMeans Clustering	-	14.7059	-	-	-	-
EM Clustering	-	17.6471	-	-	-	-
MakeDensityBasedClusterer	-	47.6471	-	-	-	-

As we can see, it is difficult to decide on one model to choose seeing that all the results are more or less revolving around 97 % of accuracy.

At first, I enjoyed trying and testing different combinations of classification algorithms, however the best result was captured by **Stacking MetaClassifier** with combining the NaïveBayes classifier and Bayesian Network classifier with choosing naiveBayes as metaClassifier.

V. Discussions:

When the overall results of the study are examined, it is seen that the Divorce Predictors Scale developed within the scope of Gottman couples therapy can predict divorce rates by a 97 % accuracy on most tested classifiers. According to this results, it can be stated that the divorce predictors were confirmed within the scope of Turkish sampling as seen in previous published articles and can really predict dysfunctional couple behavior.

And thanks to feature subset selection, we can confirm that most of divorces are mainly related the way the couples sees things: if they had the same ideas about how marriage should be or if they know each other's anxieties as proved (feature 18, 26 and 40). Reactions of couples during discussion also gives an idea about predicting a divorce.

Finally, according to the results of this study, we can say that DPS can predict divorce. This would be beneficial for the Ministry of Family, Entities handling Social problems, and more importantly the counseling services staff working on family counseling and family therapies can use this predictors to try identify the failing relationships and try to save them before things become much worse and especially families with children who can be psychologically affected more.

Furthermore and even if it's not really related to our main problem, except that it seems interesting to point to interesting point I came across in an article which was the importance of cross validation usage during the classification process mainly to avoid overfitting problems and generalize the results to an independent data, like this we could have an estimation of how accurately a predictive model will perform in practice.

For this particular problem, large sample sizes would help mitigate some of the problems with overfitting, but as mentioned in other articles the high cost and effort of collecting observational data made it very difficult to have a large dataset composed of more than 1,000 or so participants.

Finally, This project was a sum up about the various machine learning approaches through which divorces could be predicted and have also seen different methods to measure how well a model have performed.

Conclusion

To sum up, we must state that the best machine learning algorithm for a particular problem is found empirically. By trial and error. Like what was done in this project, by evaluating a suite of algorithms of different types our problem, finding what works well.

Also, while working on this project , I‘ve got across multiple article that deal with these social issues with different manners (Identification of Promising Couples using Machine Learning , Yao Liu- Chris Labiak- Matt Kliemannn), Also (The Timing of Divorce: Predicting When a Couple Will Divorce Over a 14-Year Period, john mordechai gottman - robert l wayne levenson). Which proves the significance importance of using predictive power of models to find and come with solutions related to the this social science field.

Acknowledgement : This Project was mainly based on the following research (Yöntem, M. K., Adem, K., İlhan, T. ve Kılıçarslan, S. (2019). Divorce Prediction Using Correlation Based Feature Selection and Artificial Neural Networks. *Nevşehir Hacı Bektaş Veli Üniversitesi SBE Dergisi*, 9(1), 259-273)

References

[1] https://www.kaggle.com/adisak/divorce-predictors-data-set

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/00497/

[3] https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set

[4] https://vdocuments.site/the-hazards-of-predicting-divorce-without-crossvalidation.html

[5] https://drive.google.com/file/d/0B-f7ZbfsS9-xYzlkWXYwQWFMX28/edit

[6] https://www.cs.waikato.ac.nz/ml/weka/courses.html

[7] https://ift-malta.com/wp-content/uploads/2013/01/gottman-predictor-of-divorce.pdf

[8] Brief Weka Introduction, Shuang Wu -Guided by Dr. Thanh Tran

[9] https://weka.sourceforge.io/doc.stable-3-8/weka/clusterers/MakeDensityBasedClusterer.html

[10] http://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set