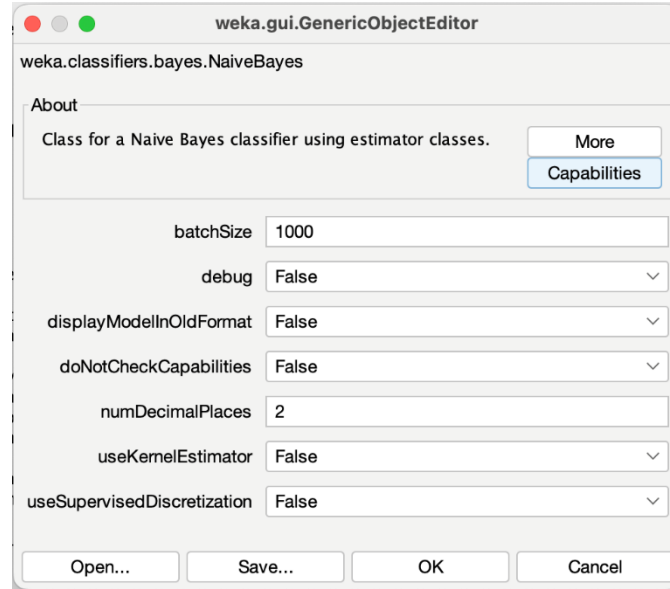


## a) Naive Bayes sınıflandırıcı.

Naive Bayes sınıflandırıcısı üzerinde sınıflandırma gerçekleştirmek için iris datasetini kullanarak Setosa, Versicolor, Virginica çiçeklerinin sınıflandırmasını amaçladım.



Naive Bayes sınıflandırması sırasında batchSize'ı 1000 olarak tercih edip ve test için 10-fold cross-validation kullanarak modeli değerlendirdim.

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances      144           96    %
Kappa statistic                    0.94
Mean absolute error                 0.0342
Root mean squared error            0.155
Relative absolute error            7.6997 %
Root relative squared error        32.8794 %
Total Number of Instances         150

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      1,000    0,000    1,000    1,000    1,000    1,000    1,000    1,000    Iris-setosa
      0,960    0,040    0,923    0,960    0,941    0,911    0,992    0,983    Iris-versicolor
      0,920    0,020    0,958    0,920    0,939    0,910    0,992    0,986    Iris-virginica
Weighted Avg.    0,960    0,020    0,960    0,960    0,960    0,940    0,994    0,989

=== Confusion Matrix ===
  a  b  c  <-- classified as
50  0  0 | a = Iris-setosa
 0 48  2 | b = Iris-versicolor
 0  4 46 | c = Iris-virginica
```

Bu sayede de sınıflar için %92 minimumda olmak üzere ortalamada %96 precision elde edebildim. İncelediğimizde F1, Recall ve Precision değerlerinin sınıflar üzerinde farklılık gösterse de ortalama değerlerde eşit sonuçlar verdiğini görebiliriz (0,96).

## b) K-ortalımalı (k-means) öbekleyici.

K-means bazlı bir öbekleme gerçekleştirmek için Amerikan seçmenlerini çeşitli evet-hayır sorularının cevabını ve hangi partiyi (Cumhuriyetçi – Demokrat) desteklediklerini içeren “vote” veri setini kullanmayı tercih ettim.

weka.gui.GenericObjectEditor

weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm. [More](#) [Capabilities](#)

canopyMaxNumCanopiesToHoldInMemory 1000

canopyMinimumCanopyDensity 2.0

canopyPeriodicPruningRate 10000

canopyT1 -1.25

canopyT2 -1.0

debug False

displayStdDevs False

distanceFunction Choose **EuclideanDistance**

doNotCheckCapabilities False

dontReplaceMissingValues False

fastDistanceCalc False

initializationMethod Random

maxIterations 500

numClusters 2

numExecutionSlots 1

preserveInstancesOrder False

reduceNumberOfDistanceCalcsViaCanopies False

seed 10

Open... Save... OK Cancel

Burada 2 seçmen grubu olduğunu baz alarak 2 cluster tercihi yaptım. Çift sayılarla cluster sayısı arttırıldığında seçmen arasında farklı alanlarla alakalı öbeklenmeler öne çıkabilir.

```
kMeans
=====

Number of iterations: 3
Within cluster sum of squared errors: 1510.0

Initial starting points (random):
Cluster 0: n,n,y,y,y,y,n,n,y,n,n,n,y,y,y, democrat
Cluster 1: n,n,y,n,y,y,y,n,n,n,n,y,n,y, democrat

Missing values globally replaced with mean/mode

Final cluster centroids:
Attribute                                Full Data      Cluster#      1
                                      (435.0)    (214.0)    (221.0)
=====
handicapped-infants                      n              n              y
water-project-cost-sharing                y              y              n
adoption-of-the-budget-resolution         y              n              y
physician-fee-freeze                     n              y              n
el-salvador-aid                          y              y              n
religious-groups-in-schools               y              y              n
anti-satellite-test-ban                  y              n              y
aid-to-nicaraguan-contras                 y              n              y
mx-missile                               y              n              y
immigration                              y              y              y
synfuels-corporation-cutback              n              n              n
education-spending                       n              y              n
superfund-right-to-sue                   y              y              n
crime                                    y              y              n
duty-free-exports                        n              n              y
export-administration-act-south-africa    y              y              y
Class                                    democrat republican democrat

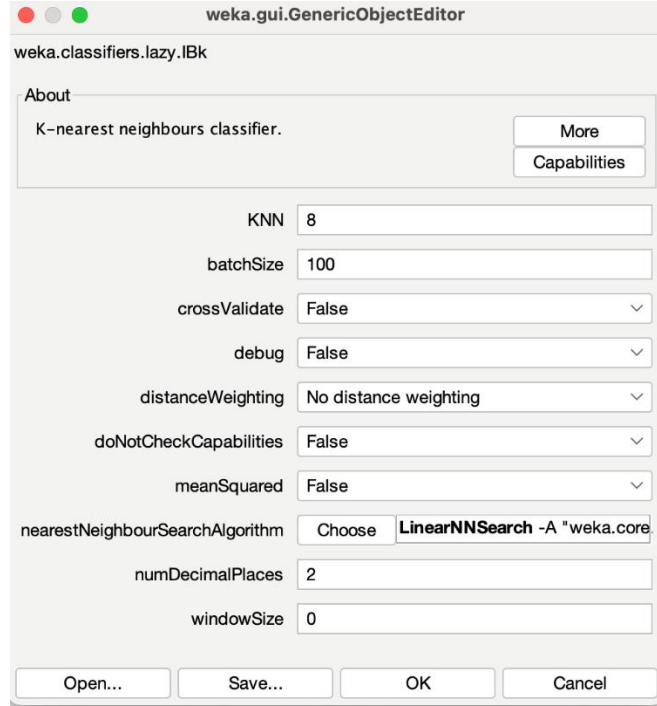
Time taken to build model (full training data) : 0 seconds

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

Sonucu incelediğimizde gördüğümüz gibi Republican ve Democrat olmak üzere 2 öbeğimiz oluşuyor

## c) k-NN sınıflandırıcı.

K-NN sınıflandırıcısında işçilerin çeşitli metrikler üzerinden değerlendirildikten sonra “iyi” veya “kötü” şeklinde sınıflandırıldığı veri seti olan “labor” veri setini kullandım.



Burada K değerini ile iteratif olarak test ederek en başarılı sonucu 8 olarak belirledim. Aynı zamanda Training için veri setinin %70'ini kullanmayı ve geri kalanı ile testleri gerçekleştirmeye karar verdim.

```
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 8 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

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

Correctly Classified Instances      51           89.4737 %
Kappa statistic                    0.7741
Mean absolute error                 0.2381
Root mean squared error             0.3127
Relative absolute error             52.0539 %
Root relative squared error         65.4946 %
Total Number of Instances          57

=== Detailed Accuracy By Class ===
               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,900    0,108    0,818     0,900    0,857     0,776    0,939    0,889    bad
               0,892    0,100    0,943     0,892    0,917     0,776    0,939    0,956    good
Weighted Avg.   0,895    0,103    0,899     0,895    0,896     0,776    0,939    0,932

=== Confusion Matrix ===
  a  b  <-- classified as
18  2  |  a = bad
 4 33  |  b = good
```

Sonuçları incelediğimizde %90'a yakın (%89) bir başarı oranıyla veri setinin test için ayrılan bölümünü sınıflandırabiliyoruz. Aynı şekilde Recall oranı da iki sınıf için yakın ve Precisiona yakın.

## d) AdaBoost sınıflandırıcı.

AdaBoost sınıflandırıcısında Tekrarlayan ve Tekrarlamayan Meme Kanserinin sınıflandırılabilmesi için oluşturulmuş “breast-cancer” verisetini kullandım.

The screenshot shows the Weka GUI with the AdaBoostM1 classifier selected. The 'Test Options' tab is active, showing 'Percentage split' at 70%. The 'Result list' on the left shows a series of training results for various classifiers, including NaiveBayes and lazy.IBk. The right pane shows the 'weka.classifiers.meta.AdaBoostM1' settings, including 'batchSize' (100), 'classifier' (DecisionStump), 'numIterations' (10), and 'weightThreshold' (100).

Eğitim esnasında veri setimin %70'ini eğitim için ayırırken test amaçlı %30'unu ayırdım. Iteration sayısı ise 10 olarak sabitlendi.

```
Weight: 0.28
Number of performed Iterations: 10

Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      63          73.2558 %
Kappa statistic                    0.3582
Mean absolute error                 0.3635
Root mean squared error             0.4493
Relative absolute error             82.5137 %
Root relative squared error         90.8425 %
Total Number of Instances          86

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
              0,944    0,625    0,718      0,944    0,816      0,407    0,684    0,765    no-recurrence-events
              0,375    0,056    0,800      0,375    0,511      0,407    0,684    0,630    recurrence-events
Weighted Avg.   0,733    0,413    0,749      0,733    0,702      0,407    0,684    0,715

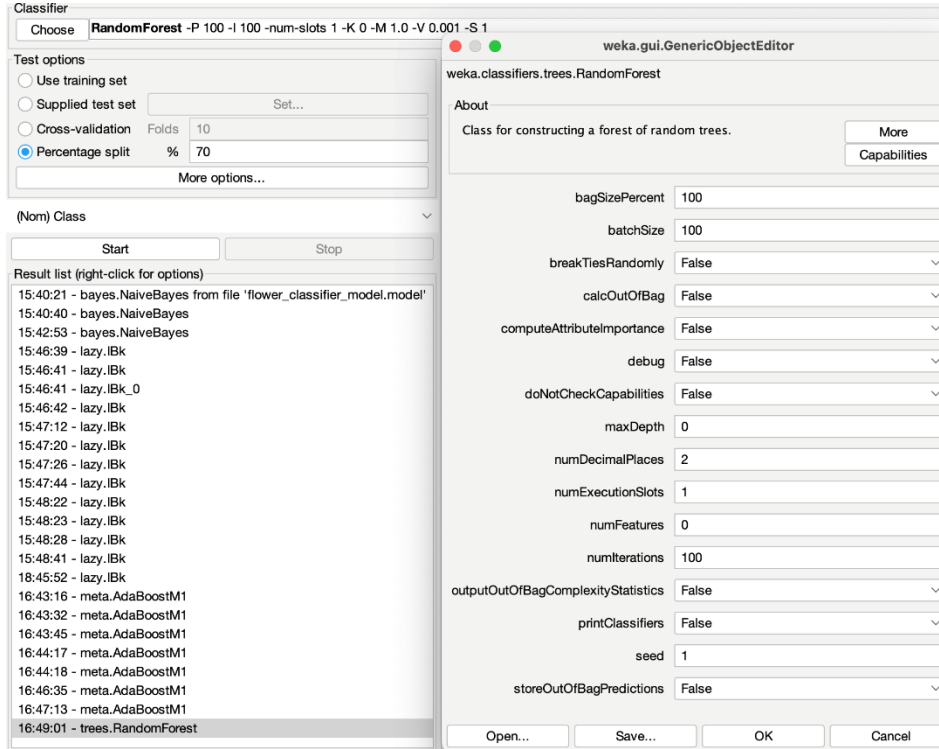
=== Confusion Matrix ===

  a  b  <-- classified as
51  3  | a = no-recurrence-events
20 12 | b = recurrence-events
```

AdaBoost diğer sınıflandırıcılara kıyasla %75 ile düşük bir sınıflandırma performansı göstermiş olsa da, veri setinin zorluğu ve model parametreleri optimizasyonu ve iterasyon eksikliği gibi birçok sebepten ötürü performans düşüklüğü oluşmuş olabilir.

## e) RandomForest sınıflandırıcı.

Random Forest sınıflandırıcısı için K-means öbekleyicisinde de kullandığım, seçmenlerin tercihlerinden oluşan “vote” veri setini kullandım.



Veri setimin %70'ini eğitim için ayırırken, 100 iterasyonda çalışmasını planladım.

```
RandomForest
Bagging with 100 iterations and base learner
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.06 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      127          97.6923 %
Kappa statistic                    0.9521
Mean absolute error                 0.0761
Root mean squared error             0.167
Relative absolute error             15.9864 %
Root relative squared error         34.0566 %
Total Number of Instances          130

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
Weighted Avg.   0.977   0.022   0.977     0.977   0.977     0.952   0.996   0.996   democrat
                  0.981   0.026   0.962     0.981   0.971     0.952   0.996   0.993   republican

=== Confusion Matrix ===
 a b  <-- classified as
76 2  | a = democrat
 1 51 | b = republican
```

Sonuçları incelediğimizde ise %97.7 Precision ve Recall oranıyla sınıflandırma yüksek doğruluk sağlamayı başardı.