Laboratorijska vežba 6 – Data Mining

Cilj vežbe: Upoznavanje sa algoritmima za klasifikaciju, regresiju i klasterizaciju i normalizacija podataka korišćenjem Weka alata.

U okviru ove vežbe koristiće se:

- Weka – Java biblioteka za mašinsko učenje,
- iris.arff – podaci za klasterizaciju,
- housing.arff – podaci za regresiju
- weather.numeric.arff – podaci za klasterizaciju

Regresija

U ovom delu vežbe biće opisan način kreiranja različitih modela za regresiju korišćenjem Weka alata.

Domen problema

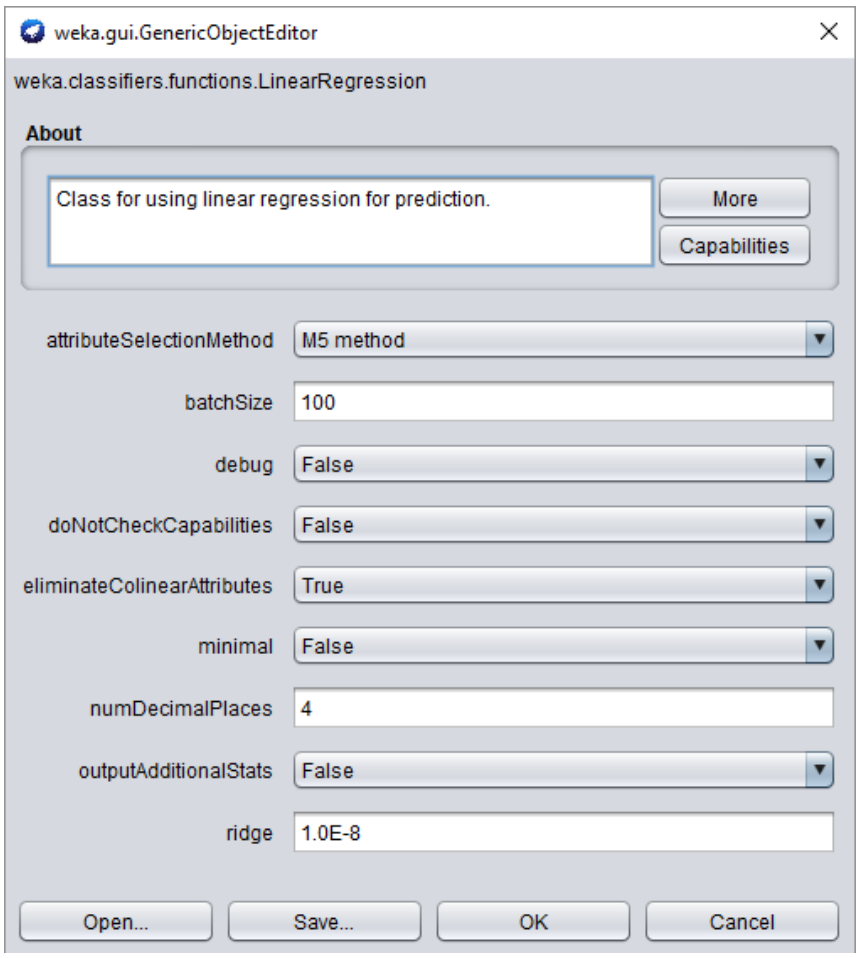
U okviru ovog dela razmatraće se problem cena nekretnina na tržištu. Potrebno je napraviti model koji određuje cenu nekretnine na osnovu njenih karakteristika. Svaka nakretnina opisana je sa 13 atributa u zavisnosti od kojih se određuje cena nekretnine.

Korišćeni podaci

Podaci koji se koriste u ovom primeru nalaze se u **housing.arff** fajlu. Postoji ukupno 506 primera.

Zadatak 1 – Linearna regresija

1. Pokrenuti Weku.
2. Po otvaranju grafičkog interfejsa, odabrati opciju **Explorer.**
3. Po otvaranju **Explorer** prozora, aktivna je **Preprocess** kartica za učitavanje i predprocesiranje podataka. Odabrati opciju **Open file** i izabrati housing.arff.
   U delu **Attributes** su prikazani svi atributi. Po potrebi, neki od atributa koji postoji u originalnom skupu se može ukloniti.
   U delu **Selected attributes** se prikazuju podaci o selektovanom atributu: koje su moguće vrednosti, koliko ima instanci sa svakom od mogućih vrednosti itd.
4. Odabrati karticu **Classify.**
5. U delu **Classifier,** odabrati opciju **Choose**. Ovom akcijom otvara se lista dostupnih algoritama.
6. Odabrati **weka/classifiers/functions/LinearRegression**
7. Klikom na ime algoritma otvaraju se podešavanja algoritma (Slika 1).

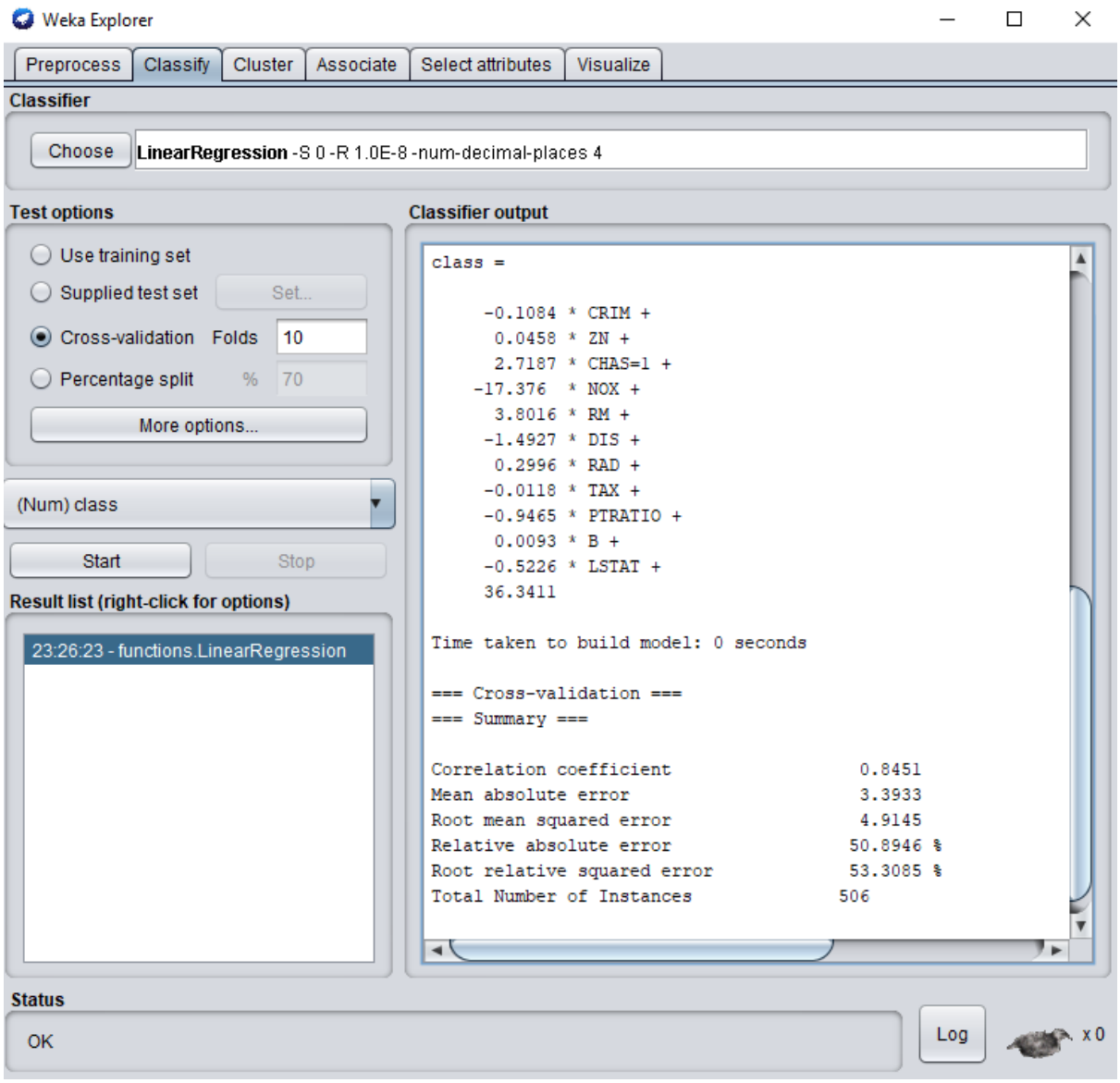


Slika 1 - Podešavanja algoritma za linearnu regresiju

Performanse ovog algoritma mogu značajno da se smanje ukoliko u dataset-u postoje atributi koji su jako korelisani. Weka ima mogućnost da prepozna i eliminiše takve atribute postavljanjem *eliminateColinearAttributes = True.*

Pored toga, aktiviranjem opcije *attributeSelectionMethod* omogućava se da Weka sama izbaci one atribute koji ne utiču na izlaz jednačine iz računice.

8. Klikom na OK zatvaraju se podešavanja algoritma.
9. U delu Test options ostaviti podrazumevane vrednosti. Za testiranje se na taj način koristi cross-validation procedura.
10. Klikom na Start izvršava se algoritam. Rezultati algoritma prikazani su na slici 2.

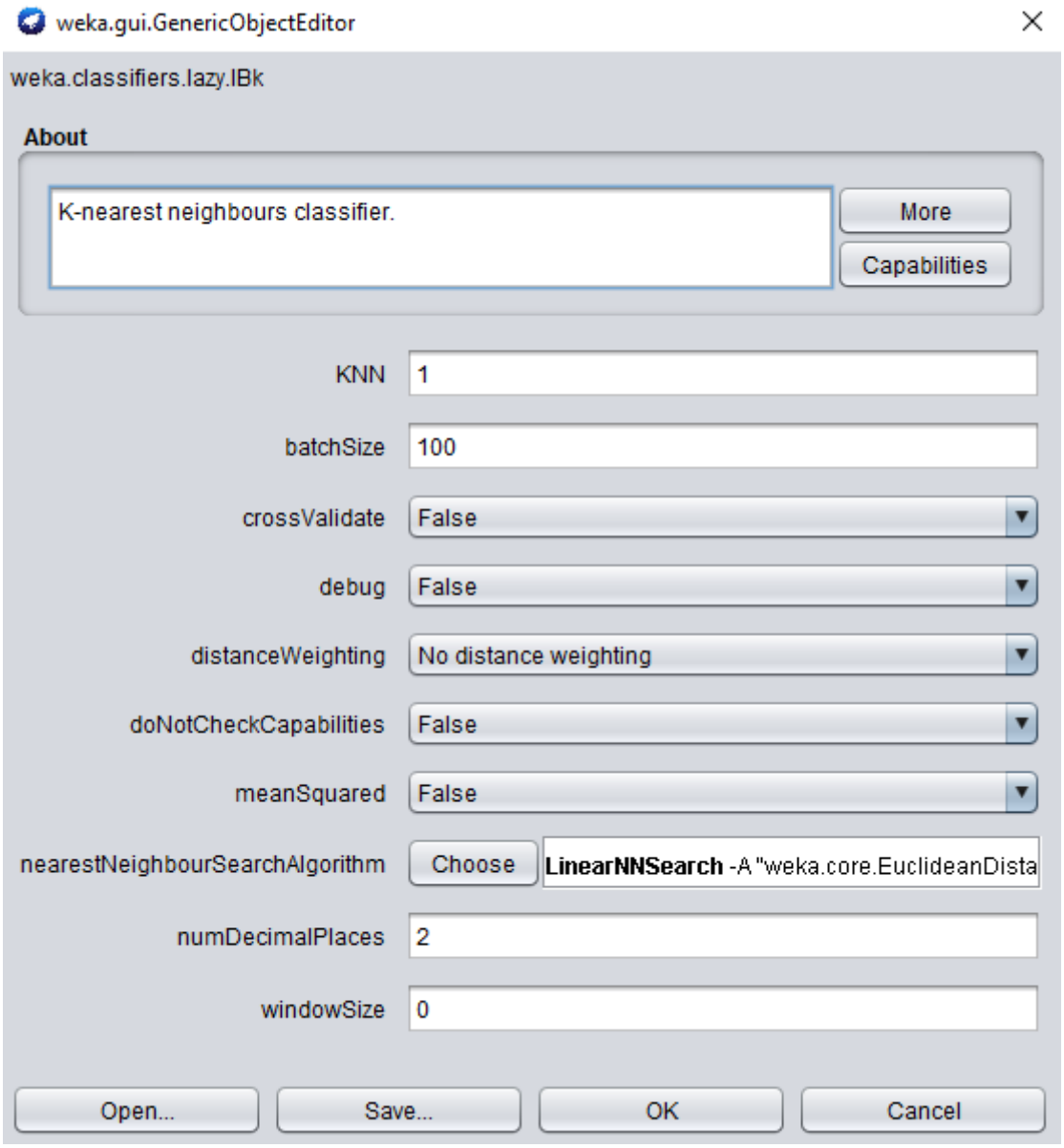


Slika 2 - Rezultati linearne regresije

U okviru rezultata se može videti dobijena linearna jednačina i rezultat testiranja modela. Pored toga, mogu da se vide neki osnovni rezultati testiranja modela i sumarne informacije. Vrednost *Correlation coefficient* uklazuje na to koliko je model pouzdan. Ovaj koeficijent može da uzme vrednost u opsegu od 0 – 1. Što je rezultat bliže 1 to je model pouzdaniji.

Zadatak 2 – k-Nearest Neighbors

1. Ponoviti korake od 1 – 5 iz prethodnog zadatka.
2. Odabrati weka/classifiers/lazy/IBk i otvoriti podešavanja algorietma
   IBk - Instance Based k

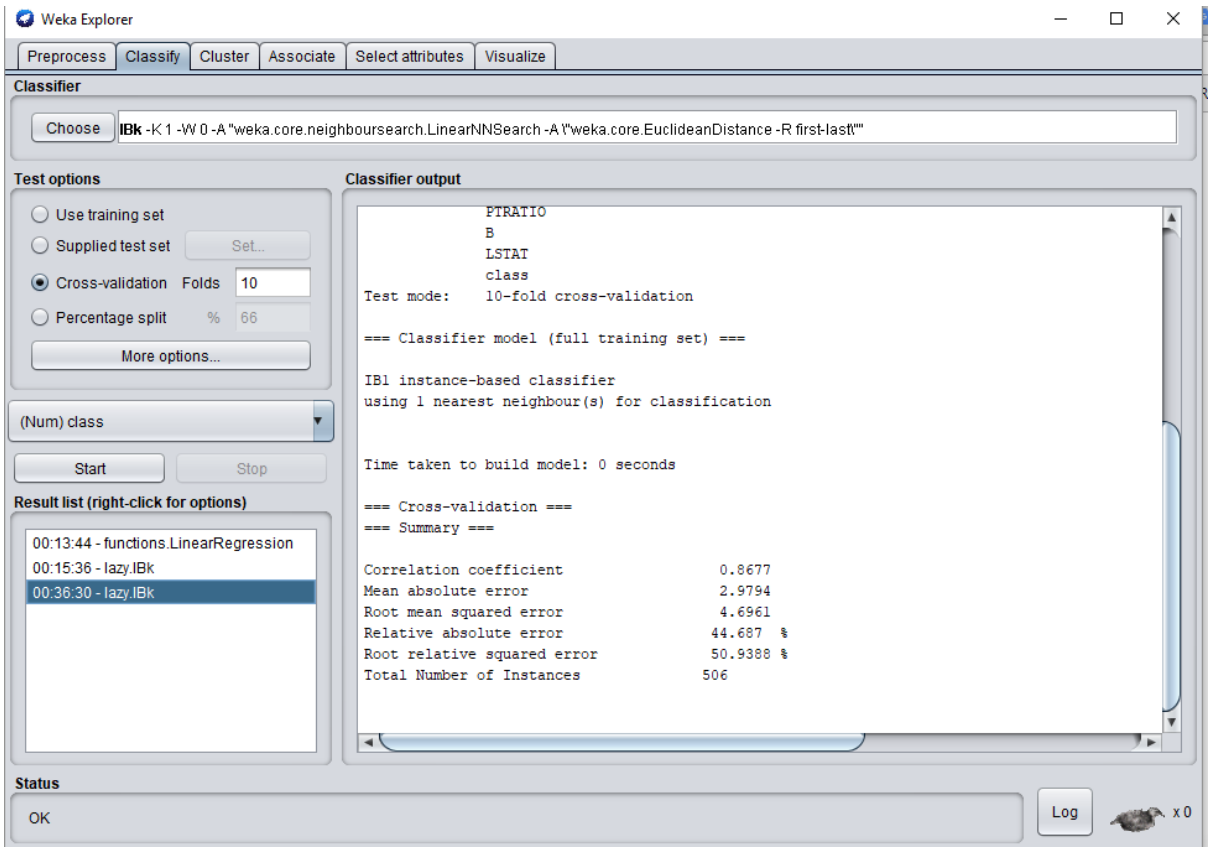


Slika 3 - IBk podešavanja algoritma

Parametar *k* određuje koliko će najbližih suseda trening set-a biti razmatrano za predikciju rezultata. Na primer, ukoliko je K = 1 za predikciju koristiće se samo jedna – najsličnija (najbliža) trening instanca, instanci za koju se traži predikcija. Uobičajene vrednosti za K su 3, 7, 11 i 21. Što je veći dataset to se uzima veća vrednost za K. Weka može sama da odredi pogodnu vrednost za K korišćenjem *cross validation* podešavanjem parametra *crossValidate* na *true*.

Drugi, jako bitan parametar prilikom podešavanja algoritma je mera udaljenosti koja se podešava korišćenjem *nearestNeighbourSearchAlgorithm* parametra. On određuje na koji način se trening set pamti i pretražuje. Osnovna vrednost je *LinearNNSearch*.

3. Klikom na OK zatvaraju se podešavanja algoritma.
4. U delu Test options ostaviti podrazumevane vrednosti. Za testiranje se na taj način koristi cross-validation procedura.
5. Klikom na Start izvršava se algoritam. Rezultati algoritma prikazani su na slici 4.



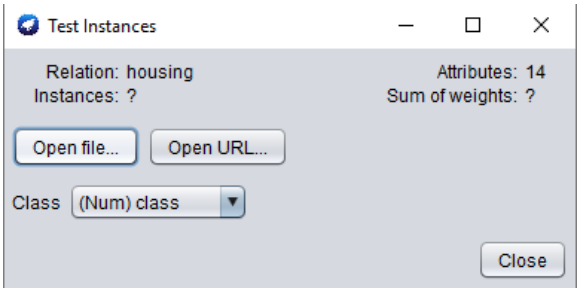
Slika 4 – Rezultati regresije korišćenjem k-Nearest Neighbors algoritma

6. Nakon kreiranja modela može da se uradi testiranje i vidi koje vrednosti bi model dodelio nekretninama za koje nisu poznate cene.
7. Kreirati *test.arff* fajl sa sadržajem:

```
@relation 'housing'
@attribute CRIM real
@attribute ZN real
@attribute INDUS real
@attribute CHAS real
@attribute NOX real
@attribute RM real
@attribute AGE real
@attribute DIS real
@attribute RAD real
@attribute TAX real
@attribute PTRATIO real
@attribute B real
@attribute LSTAT real
@attribute class real
@data
0.00632,18,2.31,0,0.538,6.575,65.2,4.09,1,296,15.3,396.9,4.98,?
0.02731,0,7.07,0,0.469,6.421,78.9,4.9671,2,242,17.8,396.9,9.14,?
0.02729,0,7.07,0,0.469,7.185,61.1,4.9671,2,242,17.8,392.83,4.03,?
```

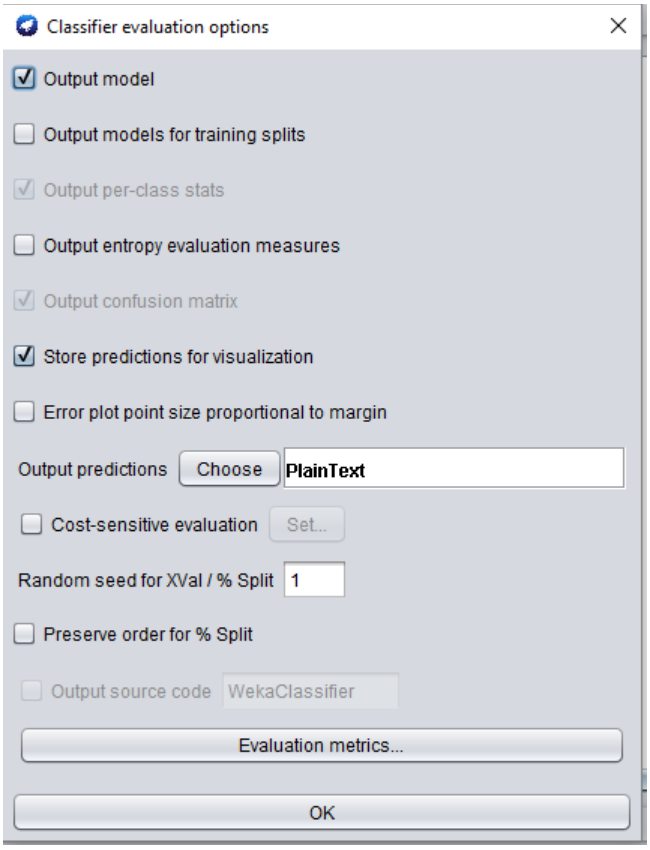
Znakovi pitanja stavljaju se na poziciju atributa **class** koji određuje cenu nekretnine koja treba da se odredi.

8. U okviru **Test options** dela izabrati **Supplied test set**. Klik na dugme **Set** otvara dijalog za učitavanje fajla sa podacima kojima treba da se odredi cena (Slika 5). Učitati kreirani test.arff fajl.



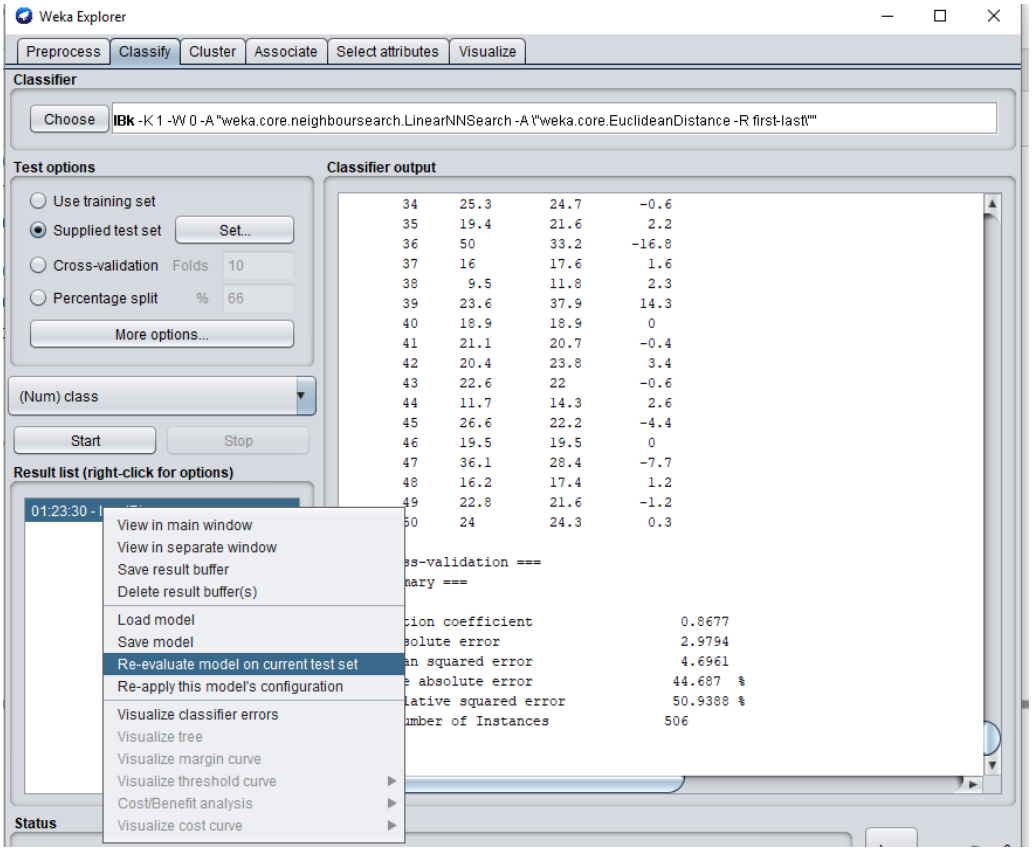
Slika 5 - Učitavanje fajla za testiranje modela

9. U okviru opcije **More options** postaviti **Output predictions** na **PlainText** (Slika 6).

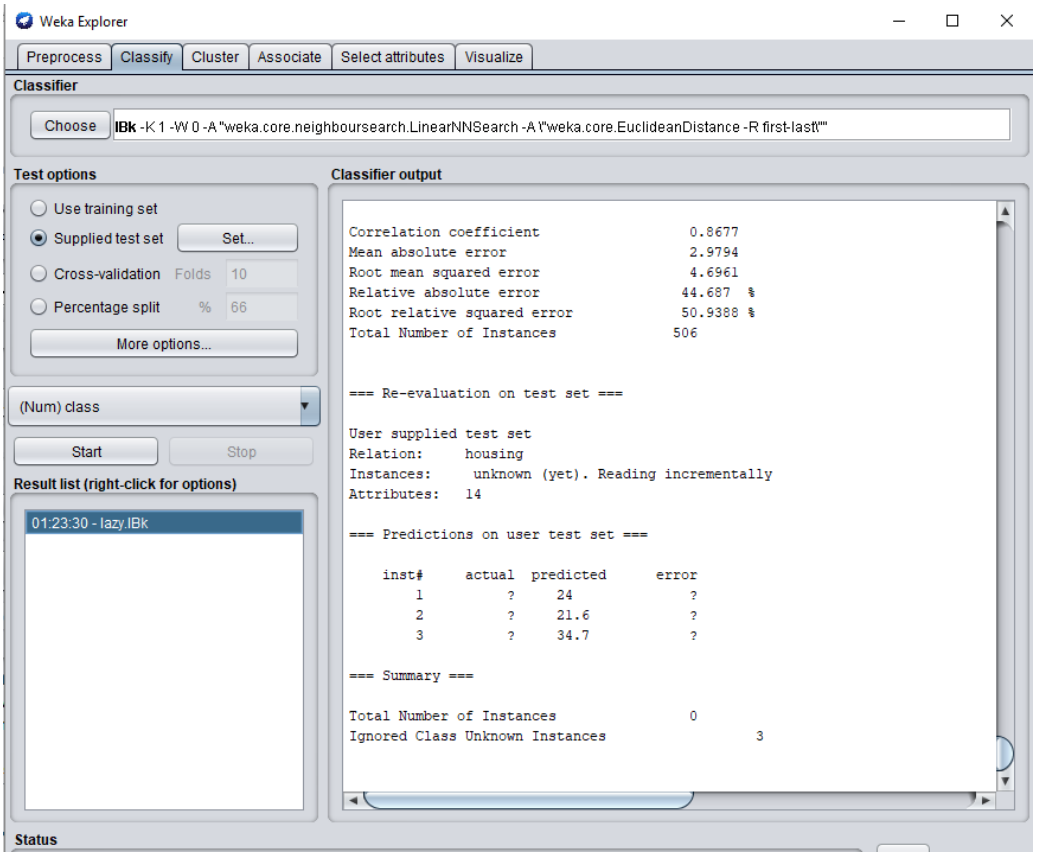


Slika 6 - Podešavanja izlaza

10. U delu menija koji se otvara desnim klikom na model odabrati opciju **Re-evaluate model on current test set** (Slika 7).
11. **Classifier output** prikazani su rezultati. Kolona **predicted** sadrži predviđene vrednosti za podatke iz test seta podataka (Slika 8).



Slika 7 - Izvršavanje modela sa test setom podataka



Slika 8 - Rezultati predikcije

Zadatak 3 – M5P algoritam

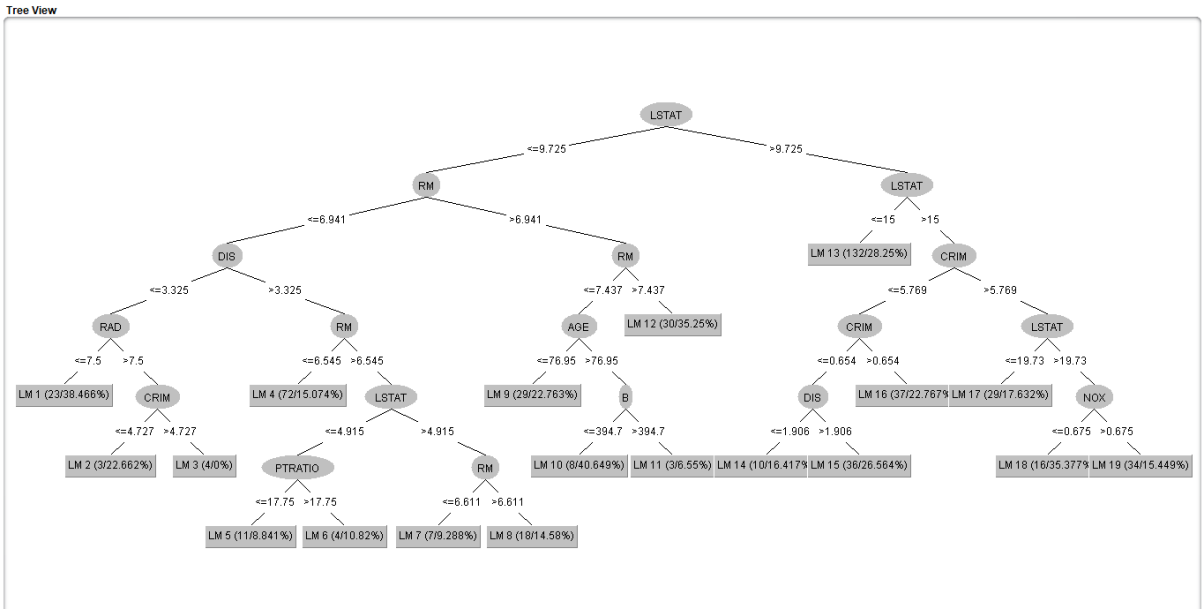
1. Ponoviti korake od 1 – 5 iz prethodnog zadatka.
2. Odabrati **weka/classifiers/trees/M5P**
3. Ostaviti podrazumevana podešavanja algoritma i podrazumevana podešavanja testiranja modela.
4. Pokrenuti algoritam.
5. Rezultati su prikazani na Slici ispod.

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9131
Mean absolute error                      2.5047
Root mean squared error                  3.7502
Relative absolute error                 37.5666 %
Root relative squared error             40.6789 %
Total Number of Instances              506
```

Slika 9 - Rezultati algoritma

6. Stablo odluke prikazano je na slici ispod.



Slika 10 - Stablno M5P algoritma

Klasterizacija

U ovom delu vežbe biće opisan način korišćenja klasterizacije u okviru Weka alata.

Domen problema

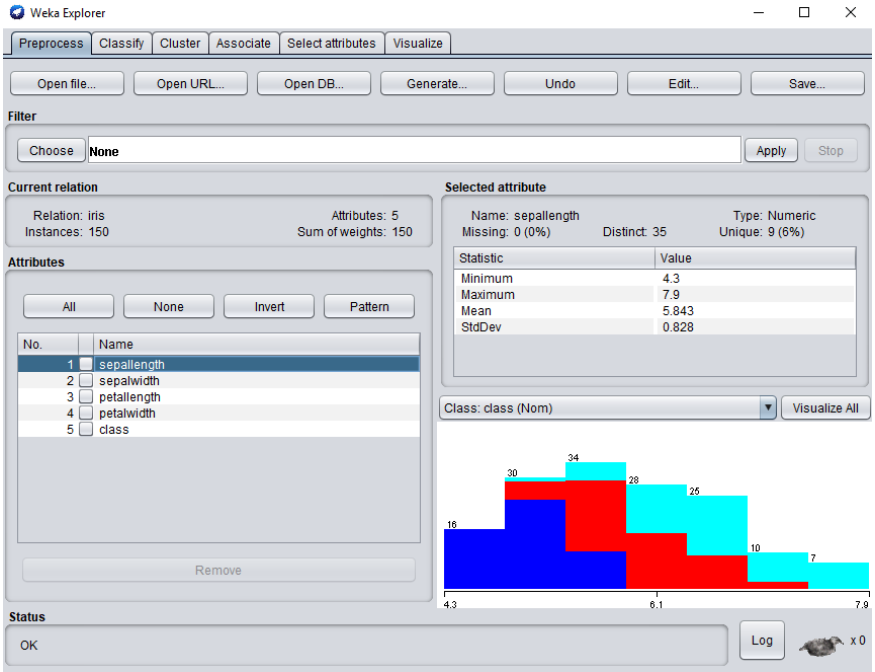
U okviru ovog dela razmatraće se klasifikacija cveća u klastere na osnovu parametra samog cveta.

Korišćeni podaci

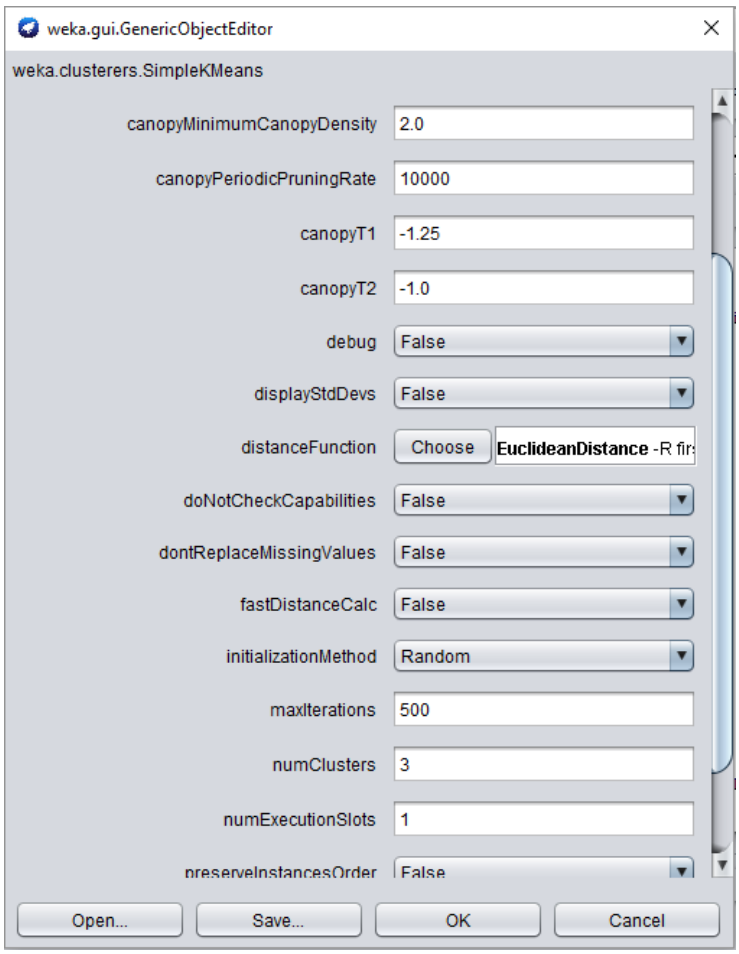
Podaci koji se koriste u ovom primeru nalaze se u **iris.arff** fajlu. Svi podaci razvrstani su u tri klase Iris Setosa, Iris Versicolour i Iris Virginica. Fajl sadrži 150 instanci (po 50 u svakoj klasi) i svaka instanca okarakterisana je sa četiri numerička atributa koja bliže opisuju parametre cveta irisa.

Zadatak 1 – Simple k-Means Clustering

1. Startovati Weka alat i učitati podatke iz fajla iris.arff (Slika 11)
2. U delu **Cluster**, odabrati opciju Choose. Ovom akcijom otvara se lista dostupnih algoritama.
3. Odabrati weka/clusters/SimpleKMeans
4. Klikom na ime algoritma otvara se prozor sa podešavanjima samog algoritma. (Slika 12)
5. Postaviti parametar *numClusters* na 3. Time se određuje na koliko klastera podaci treba da se podele.



Slika 11 - Učitavanje podataka za klasterizaciju



Slika 12 - Podešavanja algoritma

6. Klikom na OK izaći iz podešavanja
7. U okviru **Cluster Mode-a** izabrati **Classes to cluster evaluation** podešavanje u čijoj padajućoj listi treba da bude selektovano **(nom) class**. Ovim podešavanjem definiše se da se za evaluaciju klasterizacije koristi class atribut koji u učitanom skupu podataka određuje pravu podelu učitanih podataka u klase.
8. Klikom na Start izvršava se algoritam i evaluacija.
9. Rezultat algoritma prikazan je na slici 13.

```
Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4
Cluster 1: 6.2,2.9,4.3,1.3
Cluster 2: 6.9,3.1,5.1,2.3

Missing values globally replaced with mean/mode

Final cluster centroids:
                           Cluster#
Attribute      Full Data          0          1          2
                 (150.0)     (61.0)     (50.0)     (39.0)
============================================================
sepallength       5.8433     5.8885      5.006     6.8462
sepalwidth         3.054     2.7377      3.418     3.0821
petallength       3.7587     4.3967      1.464     5.7026
petalwidth        1.1987      1.418      0.244     2.0795



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

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

Clustered Instances

0       61 ( 41%)
1       50 ( 33%)
2       39 ( 26%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 47  0  3 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances :       17.0     11.3333 %
```

Slika 13 - Rezultati klasterizacije KMeans algoritmom

Zadatak 2 – Expectation Maximization algoritam

1. Startovati Weka alat i učitati podatke iz fajla iris.arff
2. U delu **Cluster**, odabrati opciju Choose. Ovom akcijom otvara se lista dostupnih algoritama.
3. Odabrati weka/clusters/EM
4. Klikom na ime algoritma otvara se prozor sa podešavanjima samog algoritma.
5. Postaviti parametar *numClusters* na 3. Time se određuje na koliko klastera podaci treba da se podele.
6. Klikom na OK izaći iz podešavanja
7. U okviru **Cluster Mode-a** izabrati **Classes to cluster evaluation** podešavanje u čijoj padajućoj listi treba da bude selektovano **(nom) class**.
8. Klikom na Start izvršava se algoritam i evaluacija.
9. Rezultat algoritma prikazan je na slici 14.

```
                (0.41)   (0.33)   (0.25)
=====================================
sepallength
  mean           5.9275    5.006   6.8085
  std. dev.      0.4817   0.3489   0.5339

sepalwidth
  mean           2.7503    3.418   3.0709
  std. dev.      0.2956   0.3772   0.2867

petallength
  mean           4.4057    1.464   5.7233
  std. dev.      0.5254   0.1718   0.4991

petalwidth
  mean           1.4131     0.244   2.1055
  std. dev.      0.2627   0.1061   0.2456



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

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

Clustered Instances

0       64 ( 43%)
1       50 ( 33%)
2       36 ( 24%)


Log likelihood: -2.055


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 50  0  0 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances :       14.0      9.3333 %
```

Slika 14 - Rezultati klasterizacije EM algoritmom

Domen problema

U okviru ovog dela razmatraće se da li je vreme pogodno za igru ili nije. Vreme je opisano parametrima izgled vremena, temperaturavlažnost i vetar.

Korišćeni podaci

Podaci koji se koriste u ovom primeru nalaze se u **weather.numeric.arff** fajlu.

Zadatak

1. Proučiti set podataka
2. Uraditi klasterizaciju za zadate podatke korišćenjem SimpleKMeans i Expectation Maximization algoritama.

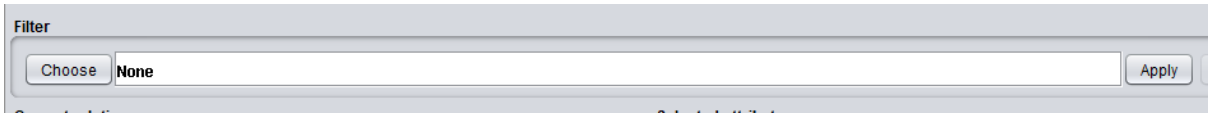
Normalizacija podataka

Za normalizaciju podataka koristi se filter Normalize koji se nalazi u okviru filtera u Preprocess kartici.

Domen problema

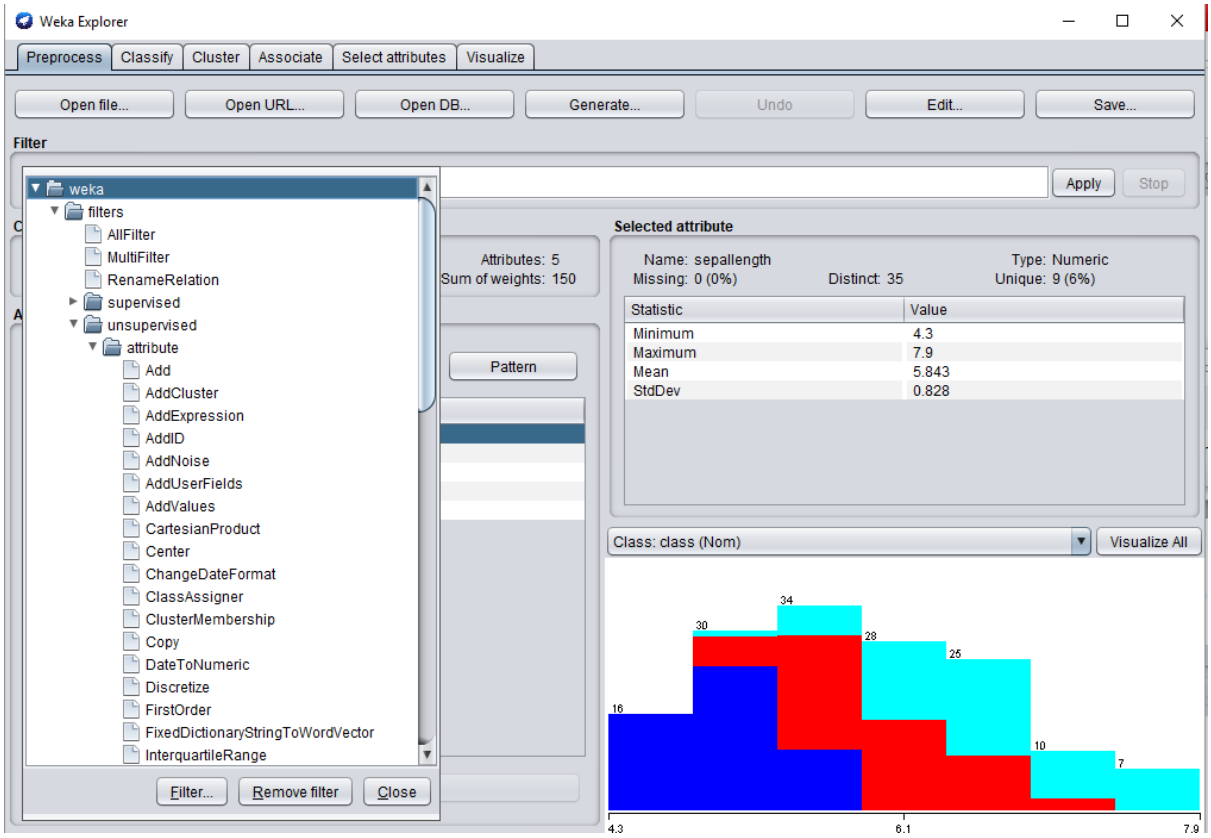
Nad podacima koji se nalaze u iris.arff fajlu isprobati klasifikaciju podataka sa normalizacijom podataka.

1. Učitati weather **iris.arff** fajl.
2. U okviru **Preprocess** kartice otvorti filtere.



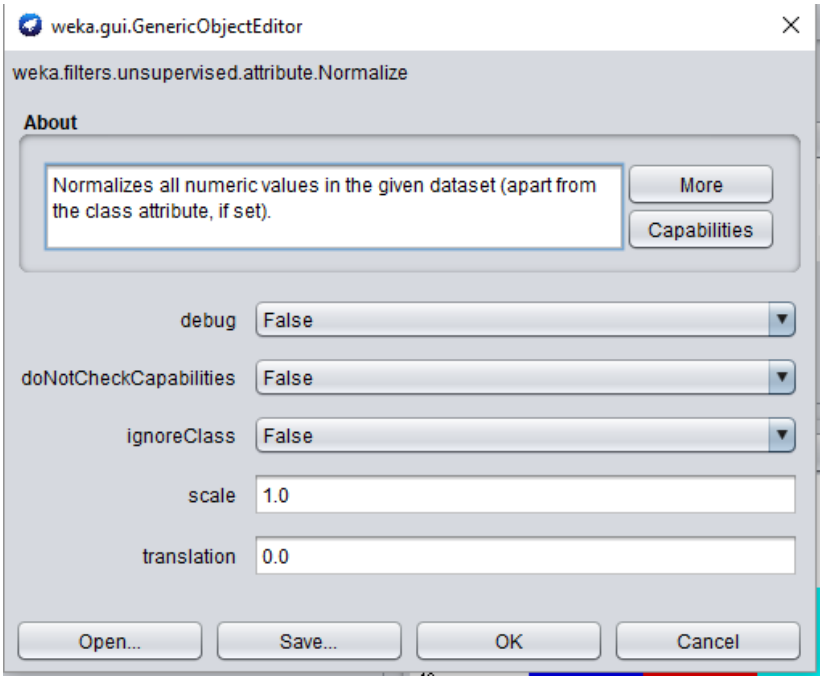
Slika 15 - Izbor filtera

3. Izabrati filter **Normalize** koji se nalazi u delu **weka/filters/unsupervised/attribute/Normalize**



Slika 16 - Izbor filtera

4. Nakon selekcije filtera otvoriti podešavanja za filter (Slika 17).



Slika 17 - Podešavanje filtera

Parametar **scale** predstavlja razliku između minimalne i maksimalne vrednosti atributa koju želimo da postavimo. Parametar **translation** označava za koliko želimo da transliramo ovaj opseg u odnosu na 0. Podrazumevana vrednost ovih parametara je 1 za scale i 0 za translate. Ovo znači da će opseg vrednosti atributa biti od 0 do 1. Ukoliko za scale stavimo 2 a translate ostane 1 opseg vrednosti biće od 0 do 2. Ukoliko scale stavimo da bude 2 a translate -1 opseg vrednosti biće od -1 do 1.

5. Nakon podešavanja scale i translate parametara izvršiti selektovani filter klikom na dugme **Apply**.
   Efekat normalizacije videće se u okviru opisa svakog od atributa ili otvaranjem podataka klikom na dugme **Edit**.
6. Model sa normalizovanim podacima zapamtiti klikom na dugme **Save**.
7. Nad ovako normalizovanim podacima isprobati IBK za različite vrednosti parametra K i NaiveBayes algoritam.
8. Uporediti rezultate.