Универзитет у крагујевцу

Факултет инжењерских наука



**Вештачка интелигенција**

Семинарски рад:

Класификација рака дојке

Студент: Предметни наставник:

Бојана Ивовић 637/2018 Весна Ранковић

Тијана Шуштершич

# Садржај

[1. Увод 3](#_Toc73357961)

[2. Визуелизација података 3](#_Toc73357962)

[3. Неуронска мрежа 6](#_Toc73357963)

[4. Коришћени алгоритми](#_Toc73357964) 6

[5. Наивни Бајес 7](#_Toc73357965)

[6. Random forest класификатор 8](#_Toc73357966)

[8. Конфузиона матрица 1](#_Toc73357967)0

[7. Закључак 11](#_Toc73357967)

[Литература 12](#_Toc73357968)

# 1. Увод

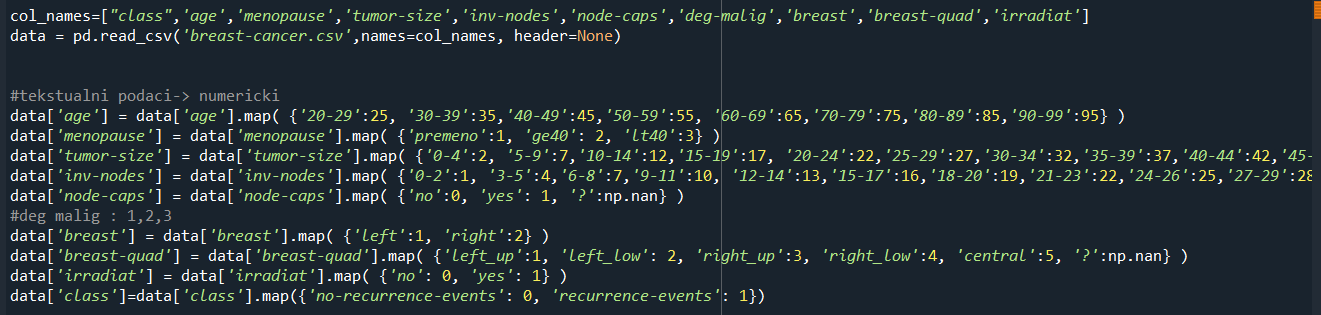
Дати су подаци везани за рак дојке који су добијени са Онколошког института у Љубљани. Од 10 колона, 9 колона представљају улазне податке, док се излазне вредности налазе у колони ‘Class’. Подаци су класификовани у две класе (no-recurrence-events, recurrence-events). Потребно је да се направи неуронска мрежа која ће на основу улазних података одређивати којој класи они припадају.

Улазних података има девет и они су дефинисани као:

* age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
* menopause: lt40, ge40, premeno.
* tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59.
* inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39.
* node-caps: yes, no.
* deg-malig: 1, 2, 3.
* breast: left, right.
* breast-quad: left-up, left-low, right-up, right-low, central.
* irradiat: yes, no.

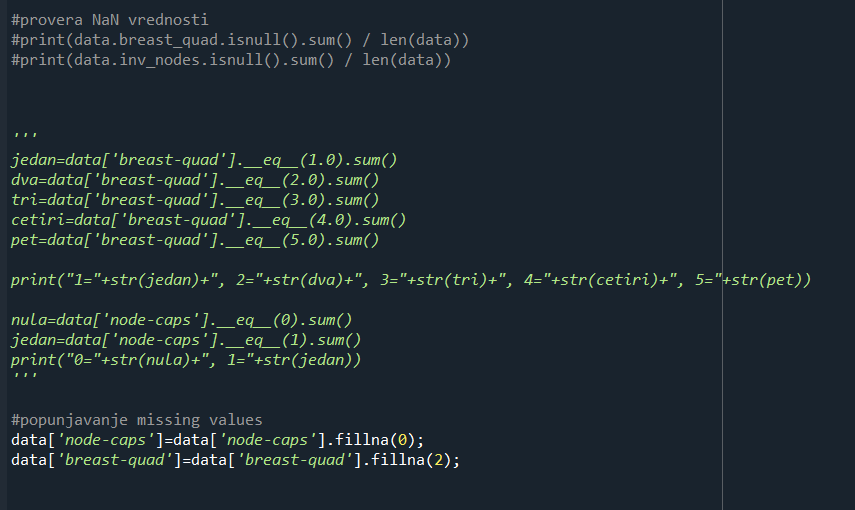
Скуп података је доступан на [1].

# 2. Визуелизација података



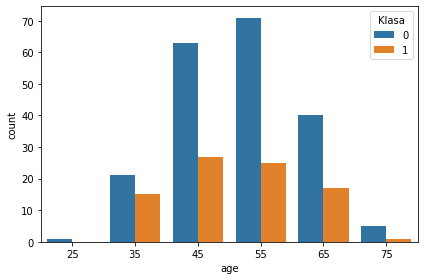
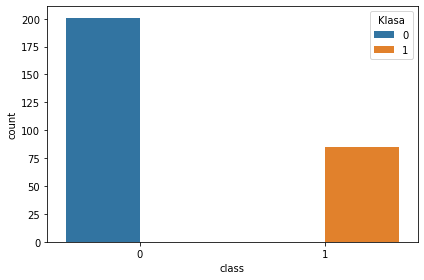
Слика1: Учитавање података и замена текстуалних података нумеричким

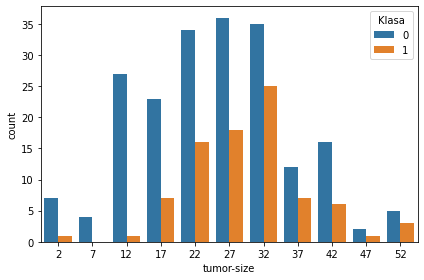
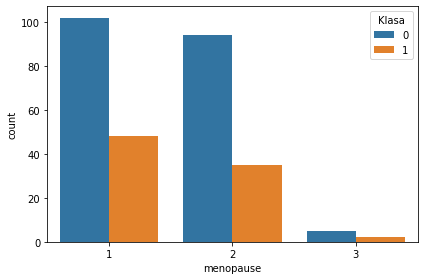
При учитавању података потребно је да сви подаци буду бројеви, тако да су сви текстуални подаци су замењени бројевним вредностима (слика1). Остали подаци остају непромењени. Примећујемо да постоје недостајуће вредности представљене знаком питања. Све такве вредности замењујемо са nan (not a number).

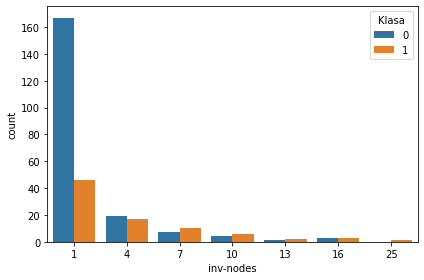
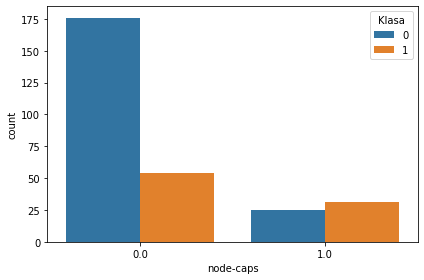


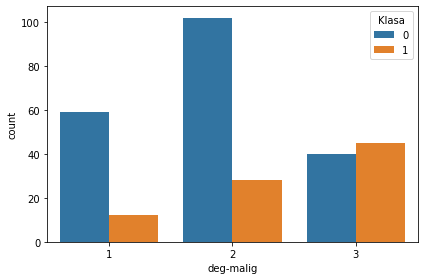
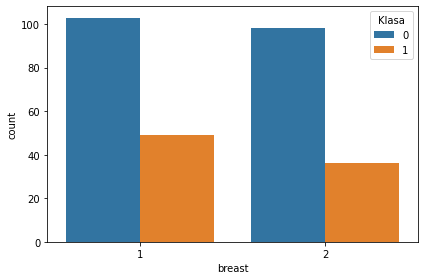
Слика2: Замена недостајућих вредности

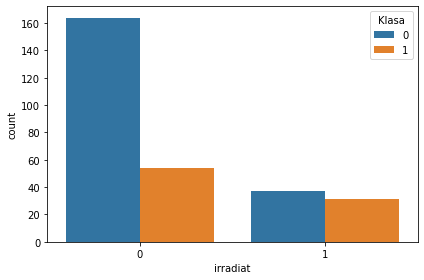
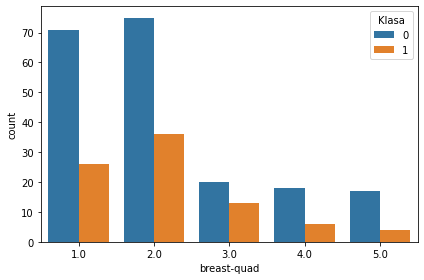
Најпре проверавамо најчешће вредности у оквиру података где се налазе недостајуће вредности (слика2). Затим их мењамо том вредношћу (нпр. “node-caps” замењујемо са 0, док “breast-quad” замењујемо са 2). Пошто су у питању категорички подаци, визуелизацију података представићемо са десет хистограма за улазне податке. За сваки податак је могуће јасно видети бројевна вредност појављивања у свакој од класа (класа 0 - no-recurrence-events , класа1 - recurrence-events).





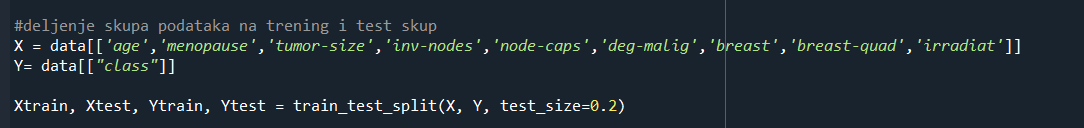






Слика3: Хистограми за класу 0 и класу 1

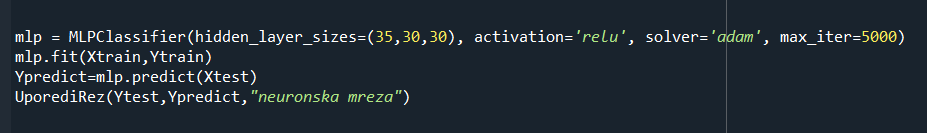
Након учитавања скупа података, потребно је поделити га на два дела - скуп за тренирање и скуп за тестирање према односу 80% према 20%. Треба обратити пажњу да су овде подаци сортирани, па их је из тог разлога потребно измешати.



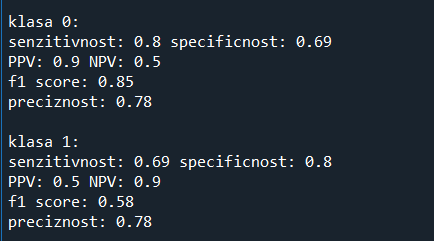
Слика4: Подела скупа података

# 3. Неуронска мрежа

За класификацију података може се користити неуронска мрежа. За израду неуронске мреже коришћен је MLPClassifier, односно вишеслојни перцептрон класификатор. Овај модел користи стохастички градијенти спуст или LBFGS да би оптимизовао функцију губитка. Овај модел тренира итераритивно, јер у сваком временском кораку израчунава працијалне изводе функције губитка у односу на параметре модела да би ажурирали параметри. За оптимизацију се користи дифолтни алгоритам ‘adam’, иако је за за мале скупове података боље користити ‘lbfgs’.



Слика5: Коришћење вишеслојног перцептрон класификатора



Слика6: Добијени резултати коришћењем MLP класификатора

# Коришћени алгоритми

За израду кода коришћен је програмски језик Python. Примењени класификатори су:

1. Gaussian naïve Bayes
2. Random forest
3. Support vector machine
4. Decision tree
5. K-Nearest neighbors
6. Logistic regression

Сваки класификатор је тестиран више пута. Најбоље резултате даје Наивни Бајесов класификатор и Random Forest. Остали класификатори дају прецизност приближну 70%.

# 5. Наивни Бајес

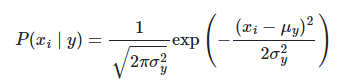
Методе наивног Бајеса су сет алгоритама за надгледано учење заснованих на Бајесовој теореми са "наивном" претпоставком о кондиционалној независности сваког пара одлика са датом вредношћу променљиве класе. Бајесова теорема, са класном променљивом y, и вектором зависности x1 до xn:



Користећи претпоставку кондиционалне независности добија се:

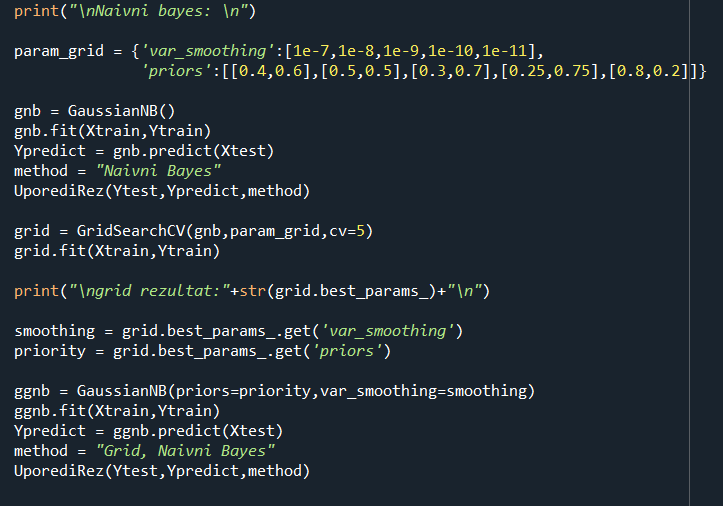


У коду је коришћен Гаусов наивни Бајес:

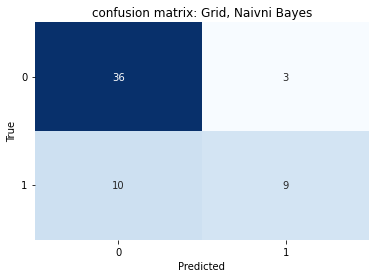


За боље резултате могуће је да се подесе grid параметри мреже. Један од хиперпараметара који постоји код овог класификатора јесте низ вредности priors који служи за измену априори вероватноћа излазних класа и користи се само онда када је скуп података неизбалансиран.

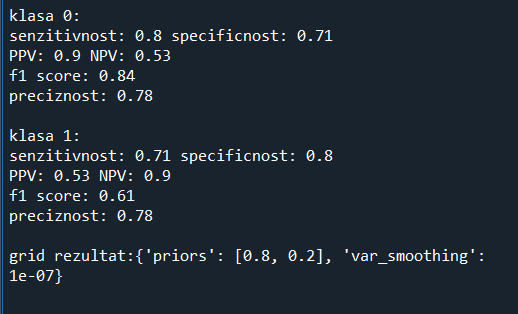
Уместо измене хиперпараметара, код наивног Бајесовог класификатора је потребно изабрати исправну методу за дефинисање модела. У овом случају се Гаусов наивни Бајес показао боље од мултиномијалног наивног Бајеса, што је и очекивано с обзиром на то да је класификација бинарна и да реални подаци често потпадају под гаусову расподелу. Прецизност је близу 80%.



Слика 7: Функција NaivniBajes



Слика8: Конфузиона матрица Наивног Бајесовог класификатора



Слика9: Добијена метрика за Наивног Бајесовог класификатора

Наивни Бајес важи за класификатор са лошим способностима генерализације јер је лако довести га и у under-fitted као и у over-fitted стање.

# 6. Random forest класификатор

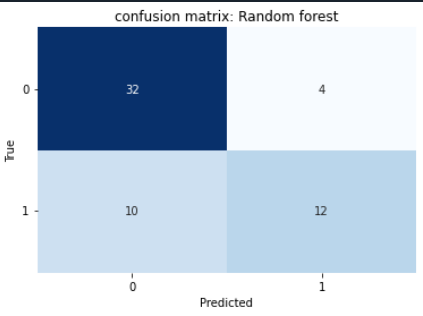
Овај класификатор ради на готово идентичан начин као стабло одлучивања. Једна од разлика је та да се уместо једног стабла, са познатим подацима, креира више стабала приближне величине. У ком ће се стаблу наћи који податак је потпуно насумично, одакле је алгоритам добио име random forest (случајне шуме). На класификатору се покреће grid search како би се пронашли оптимални параметри класификације и унапредила прецизност. Помоћу величина max\_features и min\_samples\_leaf одређује се величина узорка података који ће бити смештени у сваком стаблу. Ова оптимизација такође захтева да вредност bootstrap буде укључена. У супротном ће свако стабло имати целокупан улазни скуп података смештен у себи.

Max\_depth је параметар који ограничава највећу могућу дубину стабла. Min\_sample\_split представља минималан број одбирака улазног скупа са истом вредношћу, који је потребан за креирање новог гранања. Min\_sample\_leaf дефинише минималан број одбирака потребан за креирање листа на дну стабла, уместо гранања.

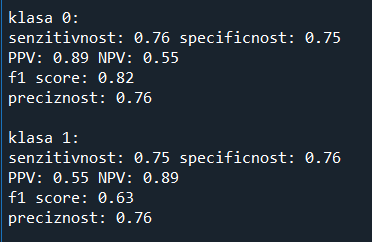
Мана овог алгоритма јесте комплкесност имплементације и време потребно за доношење одлуке.



Слика 10: Функција Random forest



Слика11: Добијена метрика за Random forest



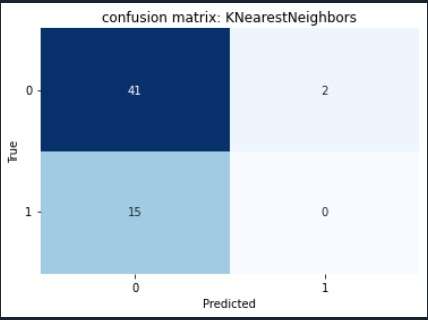
Слика12: Добијена метрика за Random forest

# 7. Конфузиона матрица

Креирамо конфузиону матрицу. Са леве стране представљене су тражене вредности излаза (TRUE), а са доње добијене вредности (PREDICTED). Из матрице може да се прочита вредноси TP (true positive), TN (true negative), FP (false positive) и FN (false negative).

* TP - класа 0 је истовремено и тражена и добијена вредност
* TN- тражена је класа различита од класе 0 и добијена је класа различита од класе 0
* FP - добијена је класа 0, али је тражена друга класа
* FN - тражена је класа 0, међутим добијена је нека друга класа

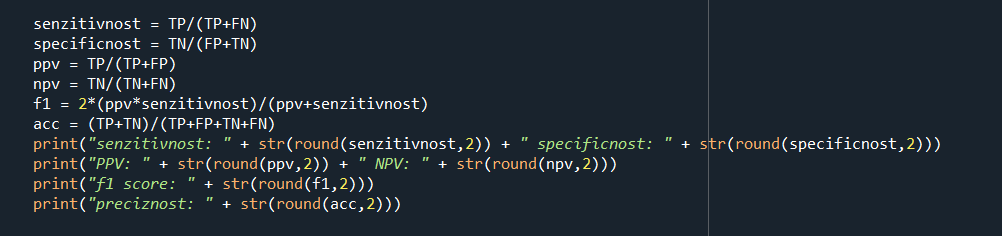
Све обрнуто такође важи за класу 1.



Слика 13: Пример конфузионе матрице

Помоћу вредности из конфузионе матрице одређују се потребне метрике.

* Сензитивност је способност класификатора да предвиђа TP за сваку класу
* Специфичност представља способност класификатора да предвиђа TN за сваку класу
* PPV(positive predictive value) и NPV(negative predictive value) за једну класу представљају удео тачних резултата у укупном броју резултата када је на излазу тражена класа (PPV) или није тражена та класа (NPV)
* Прецизност је метрика која приказује удео TP и TN у укупном броју резултата
* F1-score додатна метрика прецизности која показује однос између TP и FP, као и однос између TP и FN



Слика 14: Израчунавање метрика класификатора

# 8. Закључак

При решавању проблема потребно је прво анализирати скуп података, пре одабира алгоритма помоћу ког ће се проблем решавати. Под анализирањем података подразумева се величина целог скупа, одређивање односа тренинг и тест скупова, расподела података, број могућих улаза и број могућих класа излаза, недостајући подаци и слично.

Некада није могуће одмах одредити који алгоритам ће имати најбоље резултате, само из анализирања скупа података, већ је потребно тестирати више алгоритама и изабрати најбољи међу њима. Такође је потребно испробати више различитих комбинација хиперпараметара како би се прецизност додатно повећала у случају да почетни параметри нису најбољи.

Најбоље резултате дали су наивни Бајес и random forest. Алгоритми су добро радили и без подешавања grid параметара мреже (sklearn.GridSearchCV и sklearn.RandomSearchCV), али су подешени ради мало боље прецизности.

Најгоре резултате даје логистичка регресија. Некада даје боље резултате од осталих алгоритама, али углавном најгоре. K-nearest neighbors је мало боља од логистичке регресије, међутим њена прецизност једва достиже 70%. Остали класификатори дају приближнo добру прецизност.

# Литература

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer (15. Мај 2022. )

[2]<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> (20. Мај 2022.)

[3] https://scikit-learn.org/stable/modules/naive\_bayes.html(22. Мај 2022.)

[4] <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

(27. Мај 2022.)

[5] https://scikit-learn.org/stable/modules/neural\_networks\_supervised.html (29. Мај 2022.)