Универзитет у Крагујевцу

Факултет инжењерских наука



**Вештачка интелигенција**

Пројектни задатак:

Препознавање слова

Студент: Предметни наставник:

Тијана Танасковић 656/2017 Проф. Весна Ранковић

Садржај:

[Увод 3](#_Toc72066471)

[Визуелизација података 4](#_Toc72066472)

[Коришћени алгоритми 6](#_Toc72066473)

[1. K-Nearest neighbors 7](#_Toc72066474)

[2. Random forest 8](#_Toc72066475)

[3. Support vector machine 9](#_Toc72066476)

[Неуронска мрежа 10](#_Toc72066477)

[Добијене метрике 12](#_Toc72066478)

[Литература 15](#_Toc72066479)

# Увод

Дате су нам карактеристике скупа података слова и његови атрибути.

Подаци су дефинисани као:

-X-box [0,15]

-Y-box [0,15]

-Width [0,15]

-High [0,15]

-Onpix [0,15]

-X-bar [0,15]

-Y-bar [0,15]

-X2bar [0,15]

-Class {A,..,Z}

-Y2bar [0,15]

-Xybar [0,15]

-X2ybr [0,15]

-Xy2br [0,15]

-X-ege [0,15]

-Xegvy [0,15]

-Y-ege [0,15]

-Yegvx [0,15]

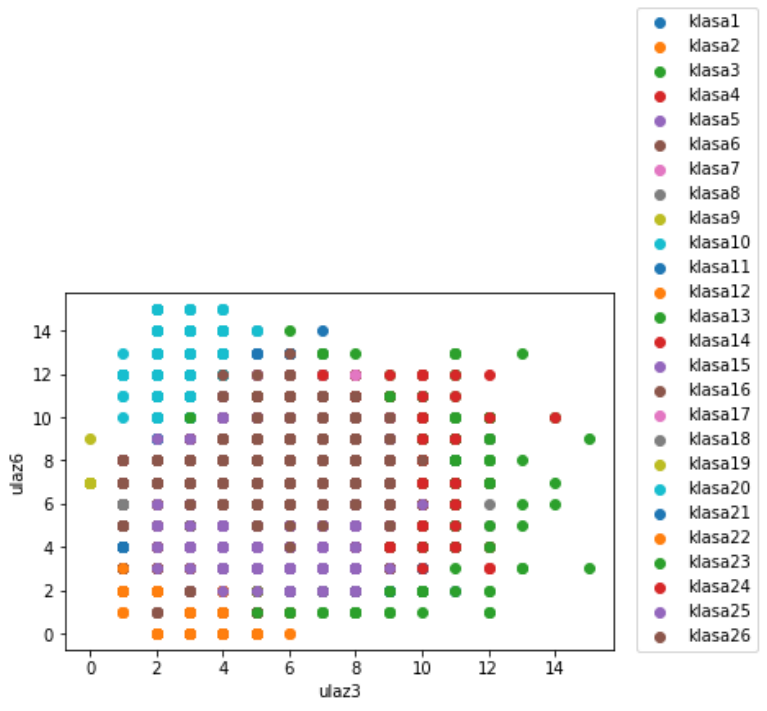
Циљ задатка је идентификовати слова енглеског алфабета на основу тога који пиксели су укључени.

# 

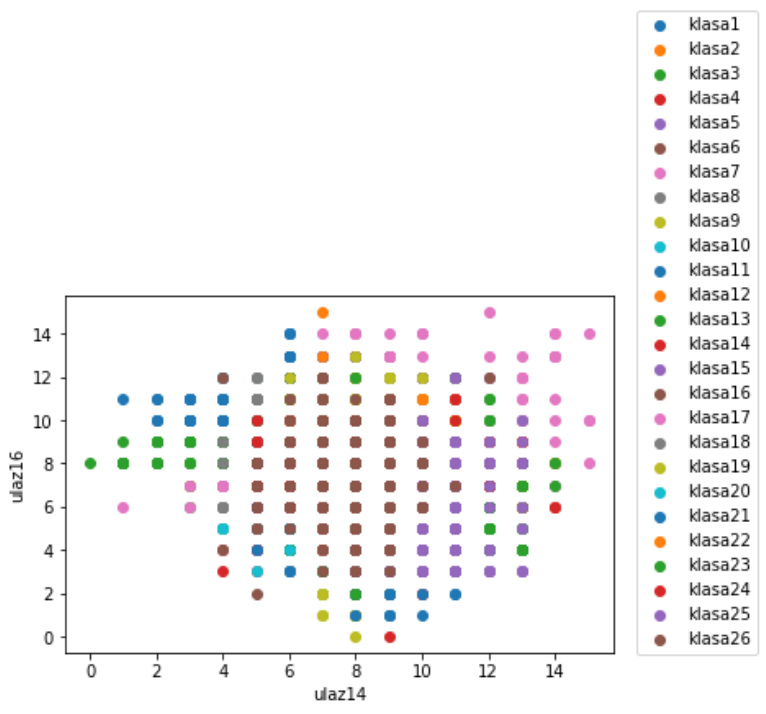
# Визуелизација података

# 

Слика 1: зависност излаза од ulaza1 и ulaza2



Слика 2: зависност излаза од ulaza3 и ulaza6



Слика 3: зависност излаза од ulaza14 и ulaza16

Подаци су подељени у 26 класа, због 26 различитих излаза, тј слова алфабета. Обојене су различитим бојама и зависности одређених улаза су приказани на претходне 3 слике.

Пошто је наш скуп података поприлично велики, претпоставка је да ће класификатор Support Vector Machine најбоље радити.

# Коришћени алгоритми

За израду семинарског рада коришћен је програмски језик Python.

Применили смо неколико класификационих алгоритама, како бисмо могли да проценимо који алгоритам најбоље ради на нашем примеру и разлику у њиховим резултатима. Коришћени класификатори су:

1. Naive Bayes
2. Stochastic Gradient Descent
3. K-Nearest neighbors
4. Decision tree
5. Random forest
6. Support Vector Machine

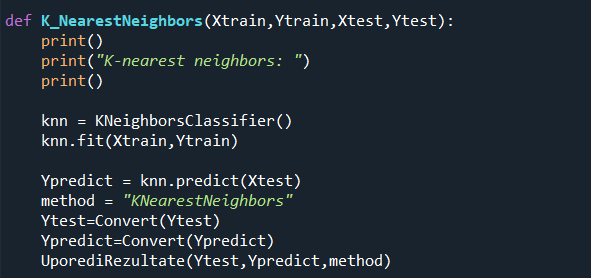
Сваки од наведених класификатора је тестиран више пута. Први класификатор (Naïve Bayes) даје баш лоше резултате, чак и до 3000 промашаја. У нашем примеру није битно која је вероватноћа да је слово А (две косе црте и једна права цртица), него нам је потребно више података да би се тај карактер запамтио. Други класификатор такође даје изузетно лоше резултате.

У даљем тексту описана су три класификатора која су дала најбоље резултате (K-Nearest neighbors, Random forest и Support Vector Machine)

Осим готових класификатора, конструисана је и неуронска мрежа у покушају достизања бољих резултата.

## K-Nearest neighbors

K-Nearest neighbors алгоритам претпоставља да су сличне ствари постоје у близини једна другој. Другим речима сличне ствари су близу. Предстаља једноставан приступ класификације где се сваки објекат класификује на основу већинске класе његових најближих суседа. Идентификовање најближих суседа врши се проналажењем удаљености између чворова у односу на све његове суседе (најчешће коришћењем еуклидске вредности).

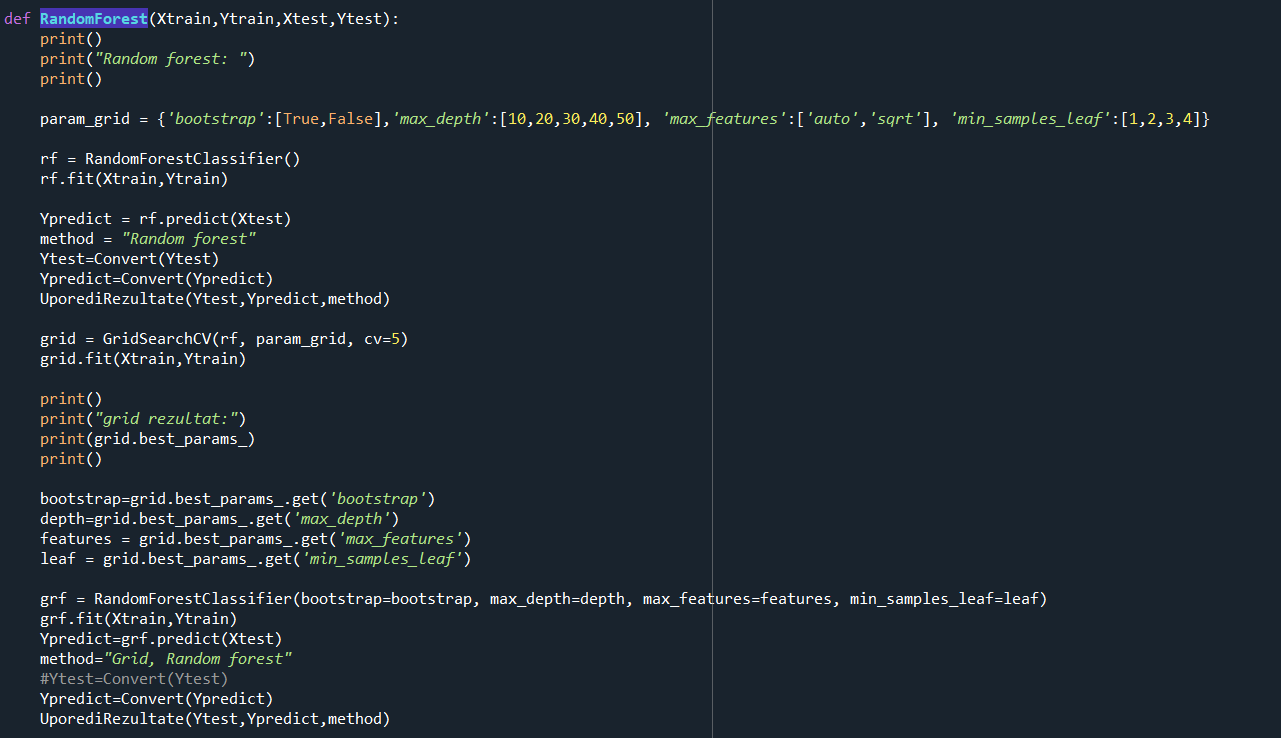


Слика 4: имплементација класификатора

## Random forest

Random forest (случајне шуме) је мета процењивач који користи већи број decision tree класификатора којима додељује разиличите делове скупа података и користи просечне вредности да унапреди прецизност предвиђања. Ова метода се користи и за класификацију и за регресију.

На класификатору покрећемо grid search како бисмо пронашли оптималне параметре класификације и унапредили прецизност. Помоћу величина *max\_features* и *min\_samples\_leaf* одређује се величина узорка података која ће бити смештена 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. Величина max\_depth представља максималну дубину стабла.



Слика 5: имплементација класификатора

## Support vector machine

Support vector machine је скуп метода за супервизирано обучавање, које се користе за решавање проблема класификације и регресије. Овај класификатор користимо због тога што је ефикасан при раду са скуповима који имају већи број димензија, што је у нашем примеру и случај.

При оптимизацији, када користимо grid search, најважнији параметар је kernel функција. Она диктира облик n-димензионалних сепаратора, помоћу којих се генерализује припадност класама. Најзаступљеније су линеарна или rbf функција.

Други параметар који посматрамо је вредност C, који представља степен до ког класификатор треба да избегне грешке, везане за погрешну класификацију тренинг података. За веће вредности C прецизност над тренинг скупом ће бити већа. Више вредности овог параметра такође захтева дуже време извршавања.



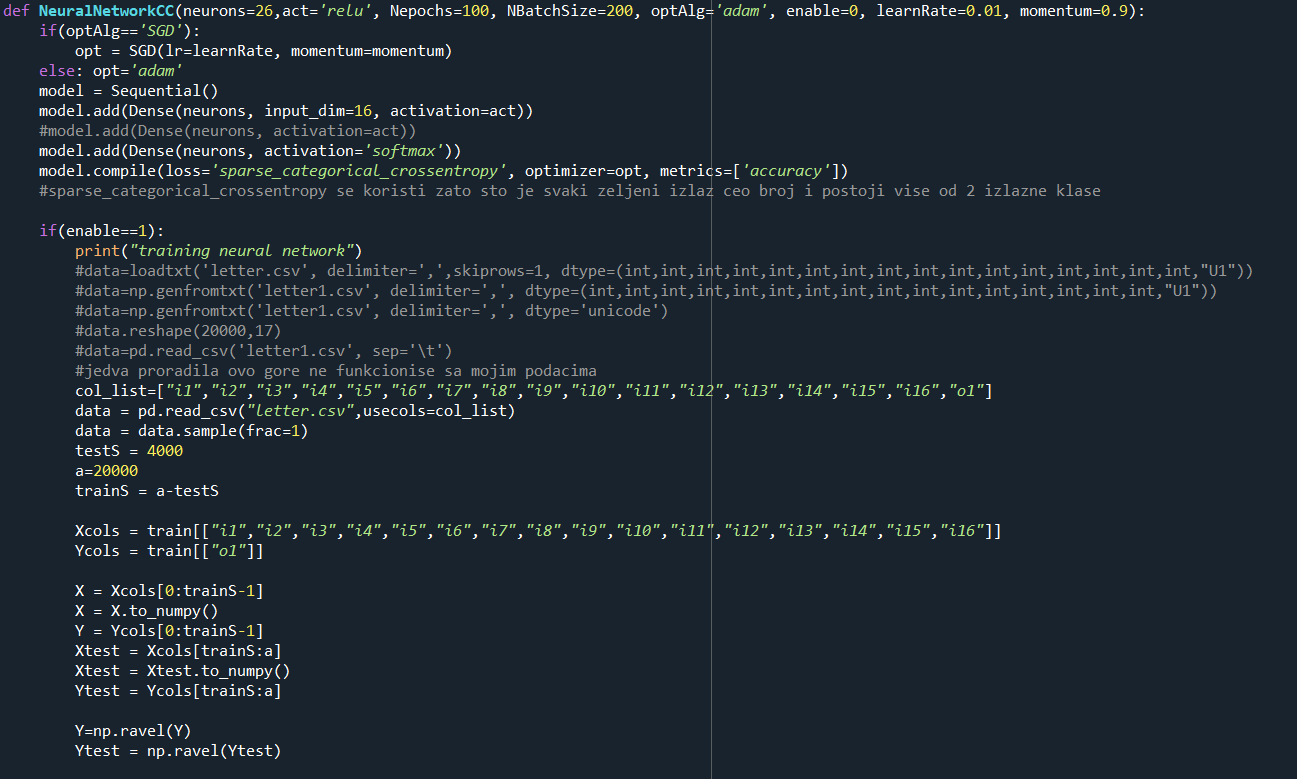
Слика 6: имплементација класификатора

# Неуронска мрежа

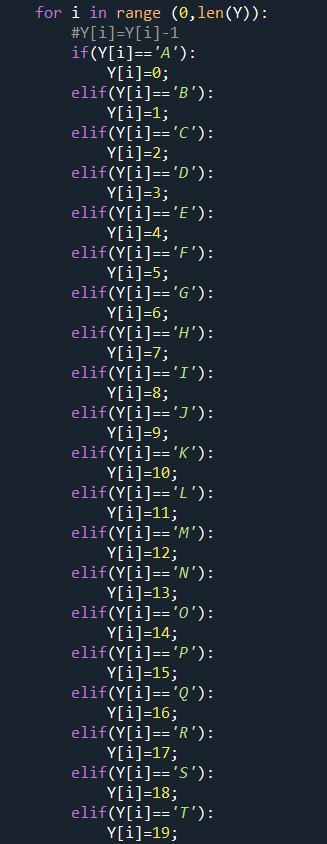
Поред готових класификатора, при изради семинарског рада коришћена је и неуронска мрежа. Хиперпараметри мреже подешавани су на основу резултата добијених коришћењем grid search алгоритма.

Мрежа има 1 скривени слој. Оба слоја имају по 26 неурона. Коришћена loss функција је *sparse\_categorical\_crossentropy.*

Обучавање се извршава у 100 епоха, где се у свакој на улаз доводи 200 одбирака. Активациона функција неурона скривеног слоја је линеарна јер она показује најбоље резултате у комбинацији са активационом функцијом излазног слоја.



Слика 7: имплементација неуронске мреже



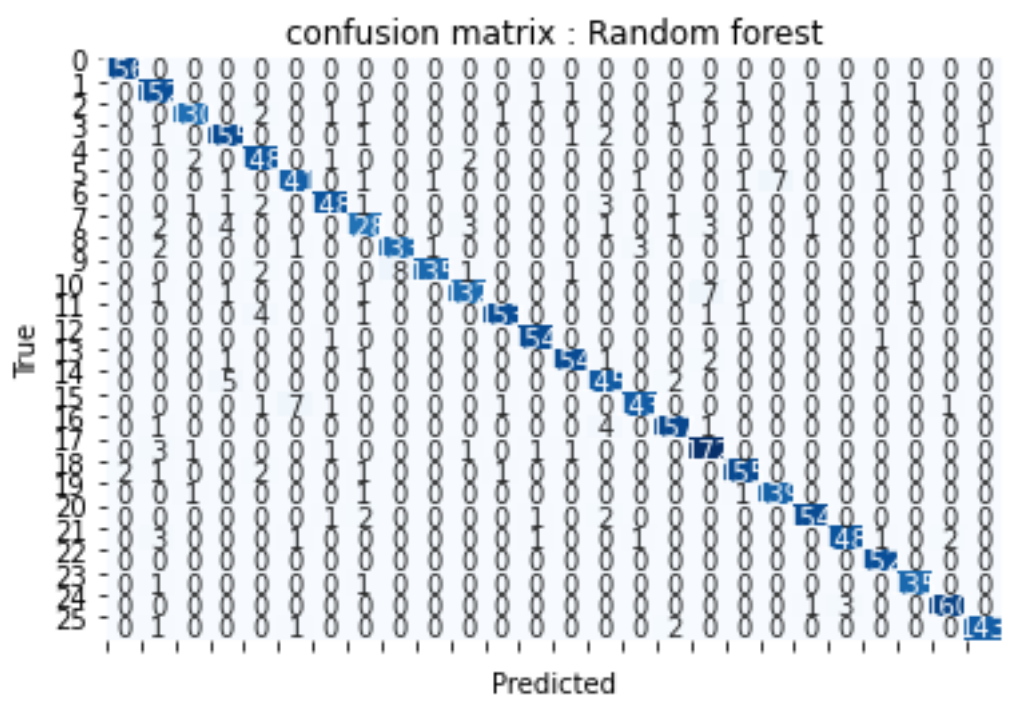
Слика 8. Словни карактери тумачени као бројеви

За оптимизацију се користи алгоритам SGD (Stochastic gradient descent), који даје боље резултате од препорученог опримизатора adam. Оптималан коефицијент брзине учења је 0.01 , а вредност гама 0.9 .

Код тренирања мреже користи се 5-fold cross валидација, односно 80% података се користи за тренирање мреже, док се осталих 20% користи за тестирање. Овај приступ је показао знатно боље резултате и за тренирање и за тестирање од 10-fold и 3-fold cross валидације.

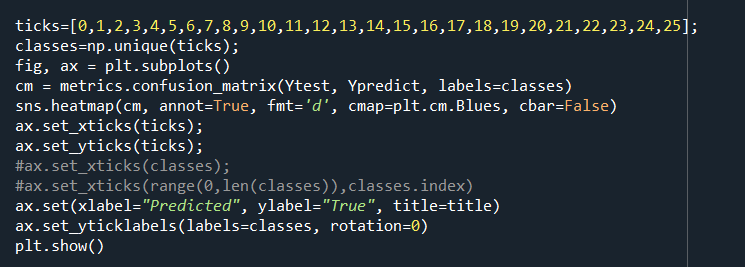
# Добијене метрике

Након извршеног тренирања и тестирања мреже, формирамо конфузиону матрицу, на основу добијених и тражених вредности.



Слика 9: конфузиона матрица добијена након примене Random forest класификатора

Конфузиона матрица је урађена само за прву класу из разлога што их постоји 26 укупно.



Слика 10: креирање конфузионе матрице

Са леве стране су представљене жељене вредности излаза, а са доње добијене реалне вредности. Матрица конфузије је битан показатељ прецизности, али и других метрика класификатора. Из матрице можемо да прочитамо вредноси TP(true positive), TN(true negative), FP(false positive) и FN(false negative).

Истинити позитивни подаци (***TP – True Positive***): када је стварна вредност позитивна, а предвиђена је такође позитивна.

Прави негатив (***TN – True Negative***): када је стварна вредност Негативна, а предвиђање је такође Негативно.

Лажно позитивни (***FP – False Positive***): Када је стварно негативно, али предвиђање је позитивно. Такође позната и као грешка типа 1

Лажно негативни (***FN – False Negative***): Када је стварни позитиван, а предвиђање негативан. Такође позната и као грешка типа 2

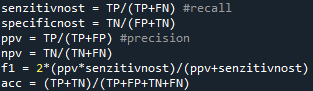
TP за класу 1 представља број пута који је мрежа на излазу добила класу 1, када је она била и жељени излаз.

TN за класу 1 представља број пута који је мрежа на излазу добила класу различиту од класе 1, када је и жељени излаз био различит од класе 1.

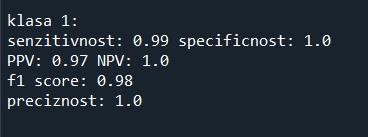
FP за класу 1 представља број пута који је мрежа на излазу добила класу 1, када она није била жељени излаз.

FN за класу 1 представља број пута које је мрежа на излазу добила класу различиту од класе 1, када је она била жељени излаз.

Помоћу ових вредности, даље, можемо да одредимо вредности метрика класификатора. Сензитивност представља способност класификатора да предвиђа TP за сваку класу, док специфичност представља способност класификатора да предвиђа TN за сваку класу [6]. PPV(positive predictive value) и NPV(negative predictive value) за класу 1 представљају удео тачних резултата у укупном броју резултата када је на излазу тражена класа 1, односно када на излазу није тражена класа 1 респективно. Тачност је метрика која приказује удео TP и TN у укупном броју резултата, а f1-score додатна метрика прецизности која показује однос између TP и FP, као и однос између TP и FN [7].



Слика 11: израчунавање метрика класификатора



Слика 12: пример добијених резултата класификације помоћу Random forest класификатора

# Литература

[1] “Veštačka inteligencija”: Moodle, Јун 2021, http://moodle.fink.rs/course/view.php?id=989

[2] “KNeighborsClassifier”: Јун 2021, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[3]”Random Forest Classifier”: Јун 2021, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[4]”Tutorial - First Neural Network”:Јун 2021, https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/