Универзитет у Крагујевцу
Факултет инжењерских наука

Семинарски рад из предмета Вештачка интелигенција

Тема:
Класификација Contraceptive Method Choice скупа података

Студент:
Никола Вуловић 570/2015

Предметни професор:
Др. Весна Ранковић
Предметни сарадник:
Тијана Шуштершић

Kragujevac 2020.

Садржај

Поставка проблема

Проблем представља креирање модела који класификује различите податке из скупа Contraceptive[1]. Скуп података се састоји од 1473 узорака из сваке од 3 класе. Задатак је да се креира модел који на основу датих атрибута разликује којој врсти података, од поменуте 3 класе, припада.

У овом случају за модел је коришћени су алгоритми SVM и PCA. Све ово је имплементирано у програмском језику Пајтон 3.8[2].

Contraceptive Method Choice data set			
Type	Classification	Origin	Real world
Features	9	(Real / Integer / Nominal)	(0 / 9 / 0)
Instances	1473	Classes	3
Missing values?			No

Слика 1 - поставка проблема

Attribute	Domain
Wife_age	[16,49]
Wife_education	[1,4]
Husband_education	[1,4]
Children	[0,16]
Wife_religion	[0,1]
Wife_working	[0,1]
Husband_occupation	[1,4]
Standard-of-living	[1,4]
Media_exposure	[0,1]
Contraceptive_method	{1,2,3}

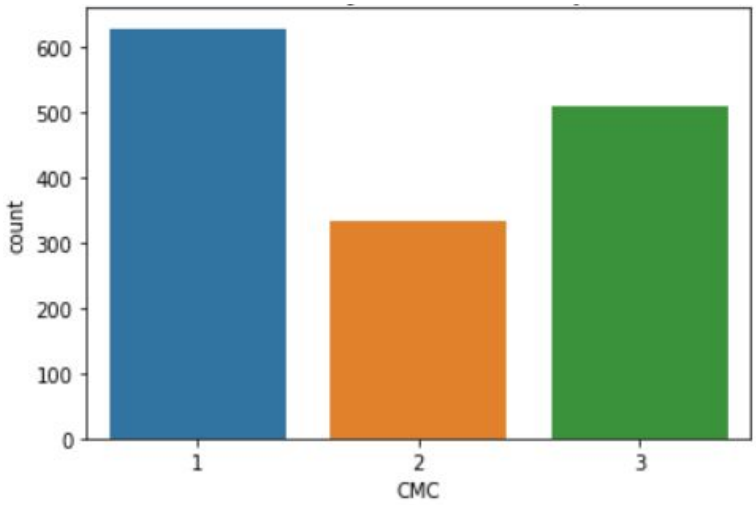
Слика 2 - улази (атрибути)

Опис и визуелизација проблема

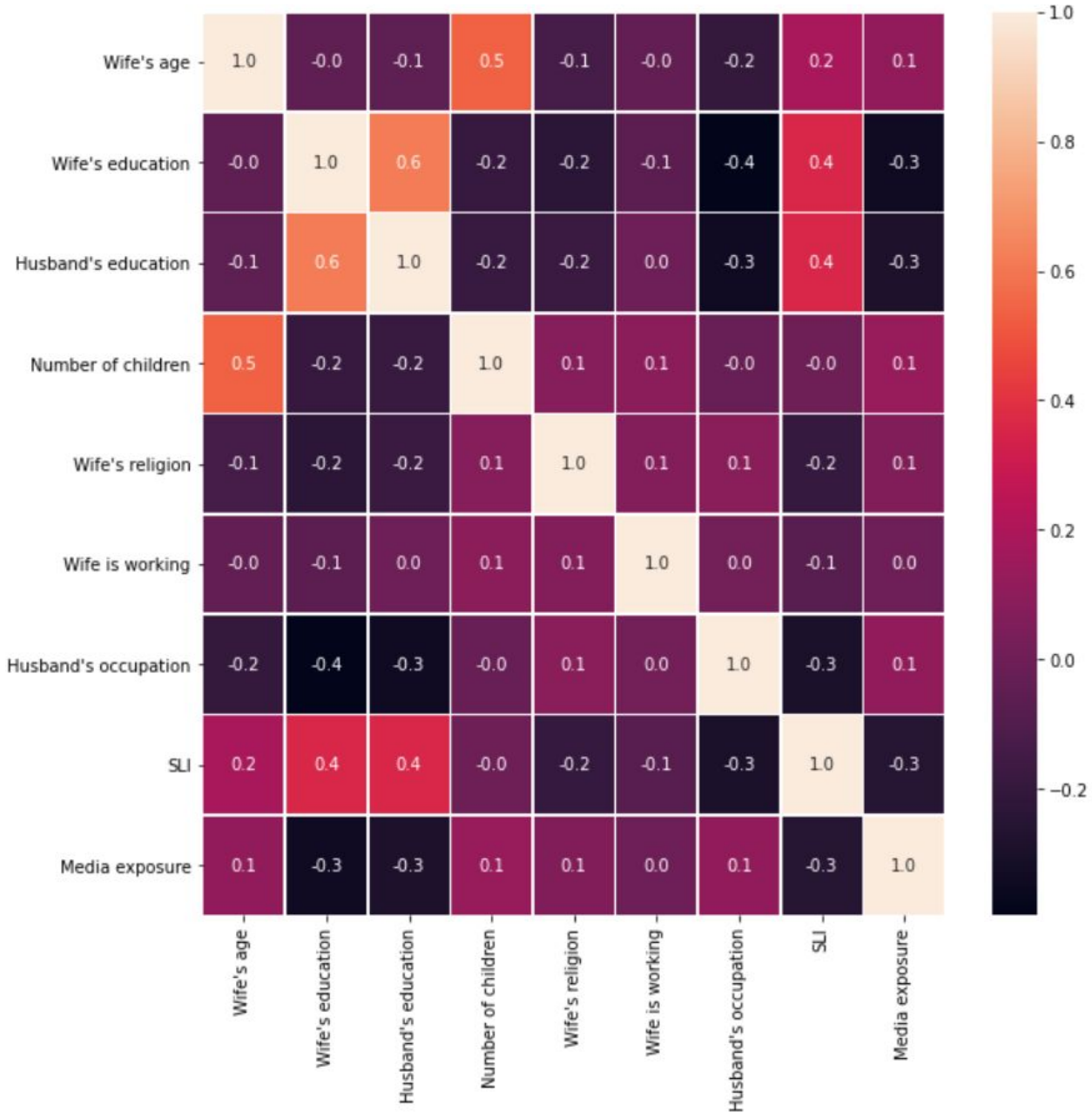
Улазне податке чине 9 атрибута (слика 2), а излазни податак представља класу којој припада узорак са тим атрибутима. Узорак може да припада једној од 3 класе. Подаци се налазе у фајлу *contraceptive.csv*. Излазне податке чини 10. колона фајла. Све вредности су нумеричке. Укупно има 1473 узорака, који су неравномерно распоређени по класама, подаци су небалансирани[3].

	Wife's age	Wife's education	Husband's education	Number of children	Wife's religion	Wife is working	Husband's occupation	SLI	Media exposure
0	24	2	3	3	1	1	2	3	0
1	45	1	3	10	1	1	3	4	0
2	43	2	3	7	1	1	3	4	0
3	42	3	2	9	1	1	3	3	0
4	36	3	3	8	1	1	3	2	0
5	19	4	4	0	1	1	3	3	0
6	38	2	3	6	1	1	3	2	0
7	21	3	3	1	1	0	3	2	0
8	27	2	3	3	1	1	3	4	0
9	45	1	1	8	1	1	2	2	1

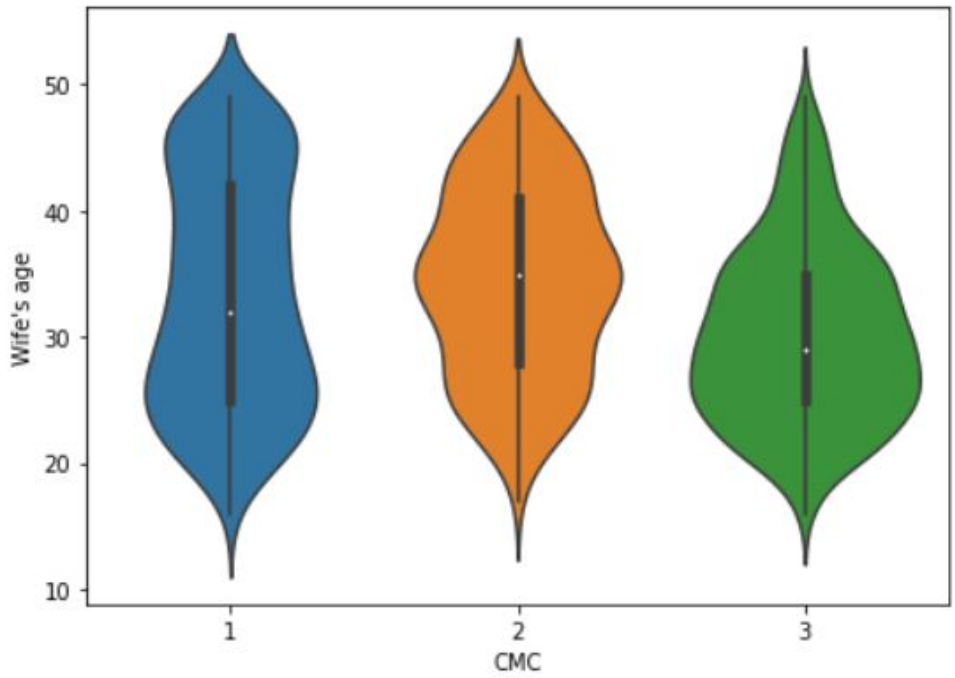
Слика 3 - Приказ првих 10 података и првих 9 атрибута



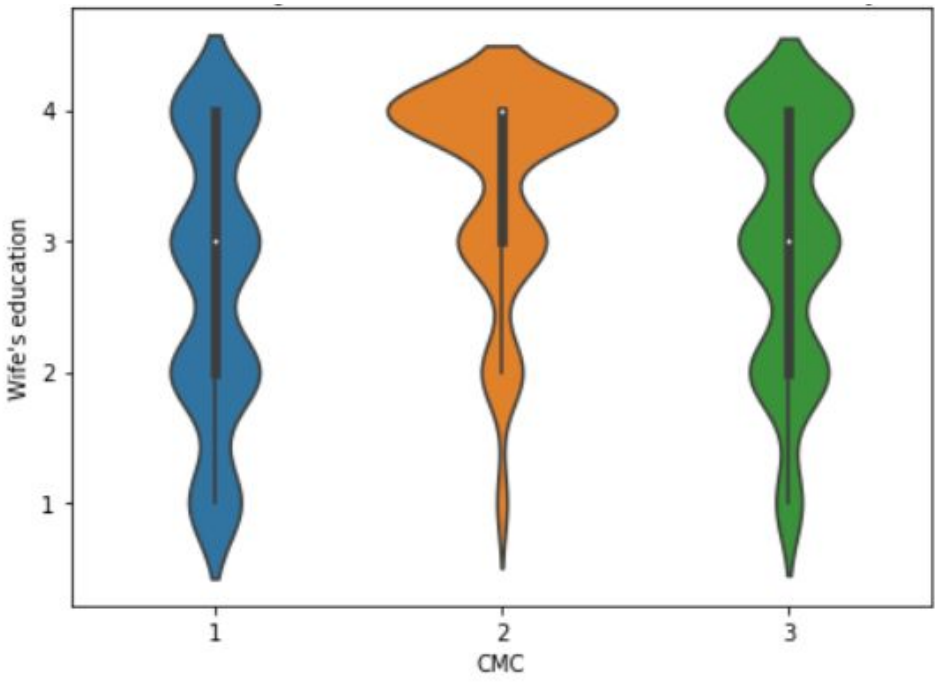
Слика 4 - Учесталост коришћења контрацептивних метода



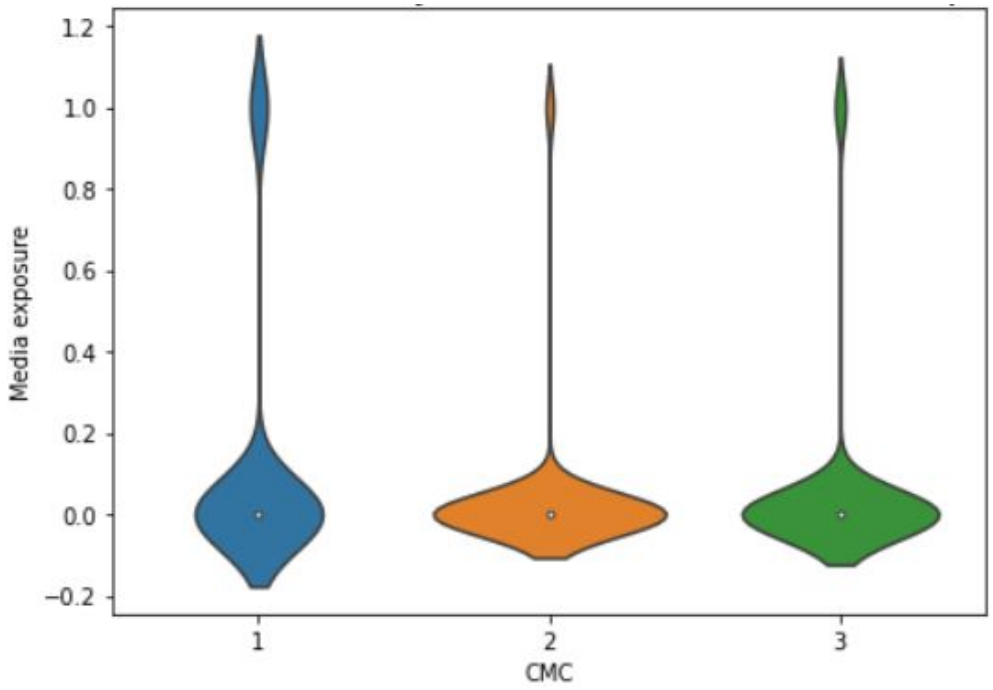
Слика 5 - Корелациона матрица контрацептивних атрибута



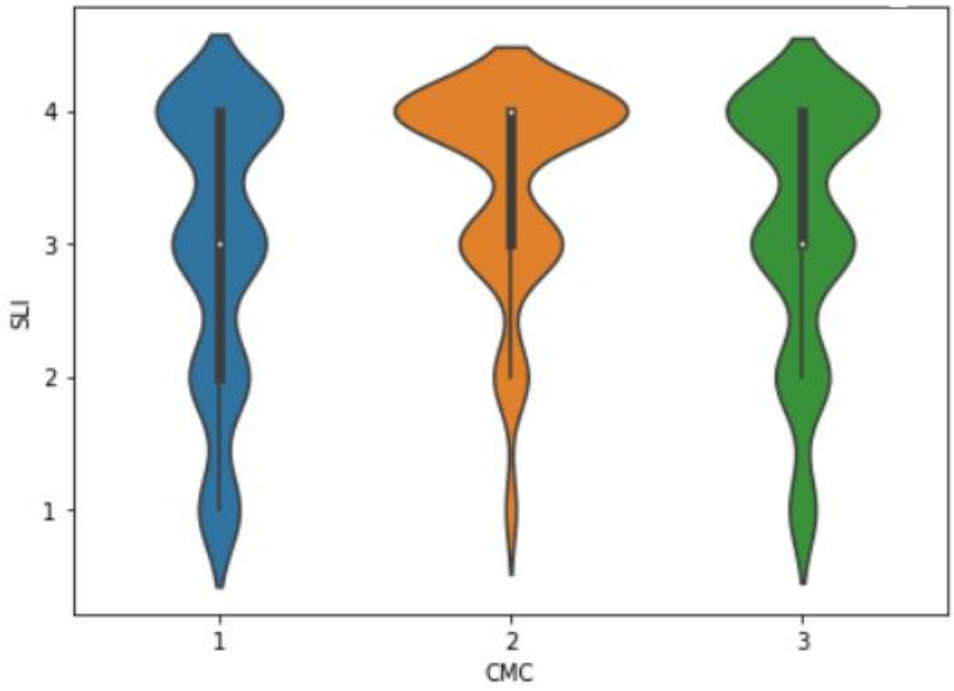
Слика 6 - Зависност година жене од избора контрацептивне методе



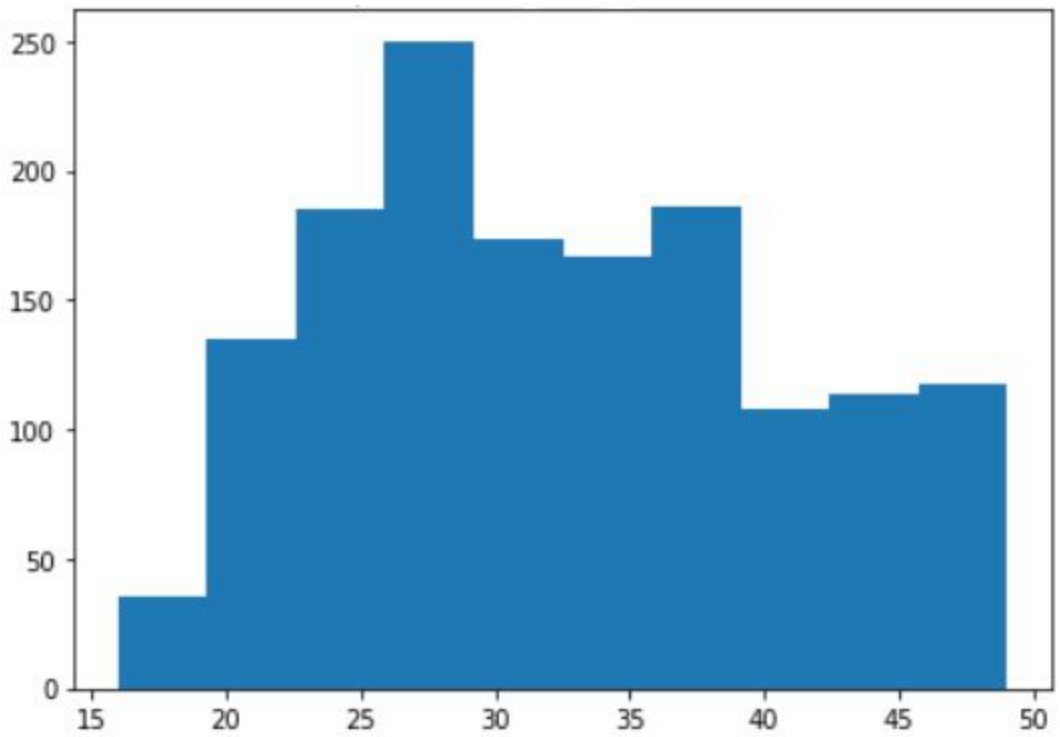
Слика 7 - Зависност броја деце од избора контрацептивне методе



Слика 8 - Зависност изложености медијима од избора контрацептивне методе



Слика 9 - Зависност животног стандарда од избора контрацептивне методе



Слика 10 - Број жена по годинама

Учитавање, раздвајање података и процесирање

Имплементација је рађена у програмском језику Пајтон 3.8[2]. Преко функције *read_csv(),* која је део *pandas* библиотеке[4], учитани су подаци из *contraceptive.csv* фајла. Улазни фајл се састоји од 10 колона. Првих 9 колона су атрибути података, док је последња колона заправо излазна колона која одређује дефинисаноист различитих класа. На слици 11 се види имплементација фунција *chi2*[5] и *SelectKBest*[6] који су део *sklearn*[7] библиотеке. Функција *SelectKBest* налази 5 атрибута са најбољим резултатом.

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
# nalaženje 5 atributa sa najboljim rezultatom
select_feature = SelectKBest(chi2, k=5).fit(X1, y1)
```

Слика 11 - Функције chi2 и SelectKBest библиотеке sklearn

Избор атрибута је техника којом у подацима одабиремо оне особине које највише доприносе циљној променљивој. Другим речима, бирамо најбоље предикторе за циљну променљиву. Класе у модулу *sklearn.feature_selection* могу се користити за одабир атрибута / смањење димензионалности на скуповима узорака, било за побољшање резултата тачности процењивача или за побољшање њихових перформанси на скуповима података са врло високим димензијама.

Chi2 тест је статистички тест хипотезе који претпоставља (нулта хипотеза) да се посматране фреквенције за категоријалну променљиву подударају са очекиваним фреквенцијама за категоријалну променљиву. Тест израчунава статистику која има хи-квадрат дистрибуцију

```
Lista rezultata: [132.68028116  45.64613752   9.48391126  45.12805438   3.22934311
   1.29943087  18.13612755  18.39981677  29.23597734]
Lista atributa: Index(['Wife's age', 'Wife's education', 'Husband's education',
       'Number of children', 'Wife's religion', 'Wife is working',
       'Husband's occupation', 'SLI', 'Media exposure'],
      dtype='object')
```

Слика 12 - Штампање резултата и атрибута

Модел, тренирање и тестирање на различитим алгоритмима

Овај проблем је најбоље решити коришћењем вишеслојне дубоке неуронске мреже. Међутим, ради презентације, коришћени су алгоритми машинског учења као што су поменути *Support-Vector Machines*(SVM) и *Principal component analysis*(PCA).

SVM, тј. *Support-Vector Machines*, су супервизирани модели учења који садрже алгоритме коришћене за класификацију и регресиону анализу. Ако дамо свакоме сет тренинг примера, где сваки припада једној или другој категорији, SVM тренинг алгоритам гради модел који приписује нове примере једној или другој категорији. Овиме креира модел који је невероватни бинарни линеарни класификатор, додуше методе као што је Платово скалирање постоје да користе SVM у окружењу класификовања са вероватноћом. SVM, поред линеарне класификације, може се применити и нелинеарну класификацију, користећи нешто што се зове кернел трик, имплицитно мапирајући улазе у вишедимензионалне просторе.

Подаци се деле у два скупа, за тренирање и за тестирање. Тренинг скуп података чини 70% података, док скуп за тестирање чини преосталих 30% података. Подела је направљена помоћу функција train_test_split() која за параметар прима улаз и излаз, проценат података за тест, као и random_state променљиву која конролише питање примењено на податке пре поделе. Функција је део библиотеке sklearn [5] и враћа улазни и излазни скуп података за тренирање, као и улазни и излазни скуп података за тест. Код за учитавање података и подешавање на тренинг скупу и тест скуп приказан је на слици 13.

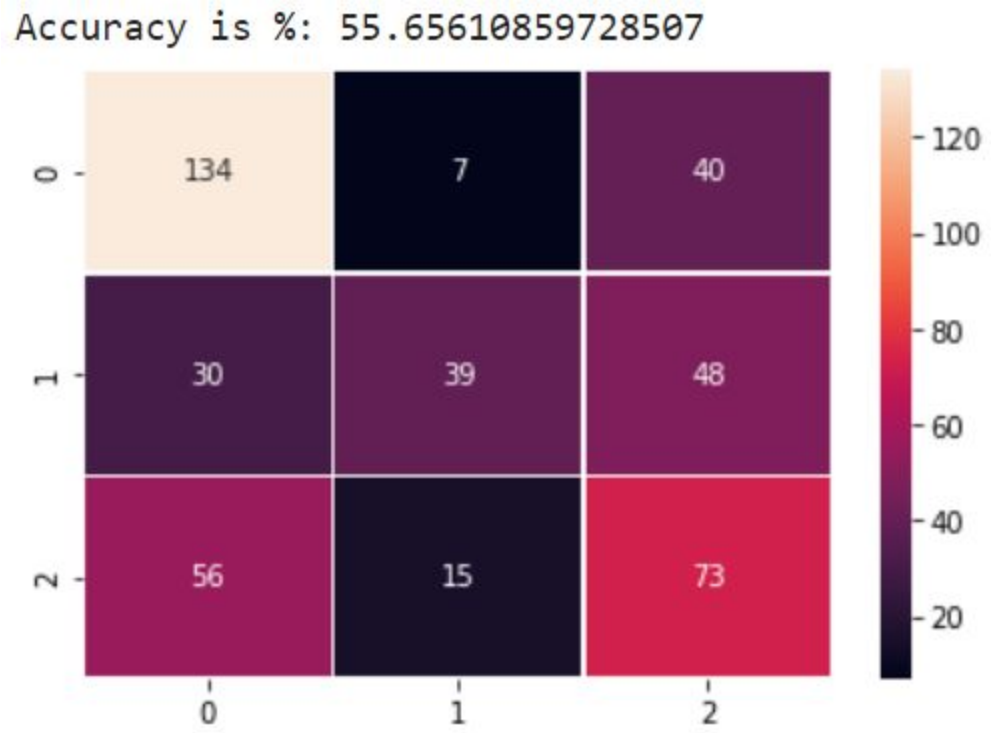
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_chi, y1, test_size = 0.3, random_state = 0)
```

Слика 13 - Примена train_test_split, као и стварање параметара SVM алгоритам

Функција fit() (слика 14) прво учитава улазне и излазне податке а затим се позивају функције accuracy_score који рачунају прецизност теста и тренинга. Прецизност теста су представљени у конфузионој матрици на слици 15.

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf',random_state = 2)
classifier.fit(X_train,y_train)
```

Слика 14 - Примена функције fit()



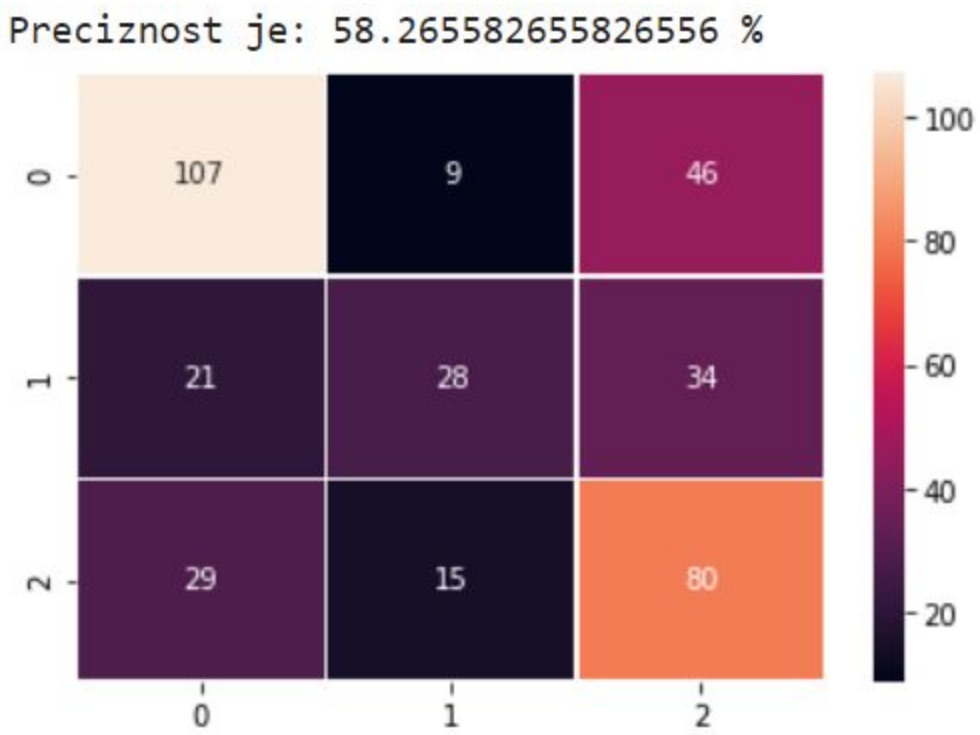
Слика 15 - Конфузиона матрица зависности резултата теста од тренинга

Principal component analysis (PCA)[8] је поступак израчунавања главних компоненти и њихове употребе за вршење промене основе на подацима, понекад

користећи само неколико првих главних компоненти, а игноришући остале. PCA се користи у истраживачкој анализи података и за израду предиктивних модела. Обично се користи за смањење димензионалности пројектовањем сваке тачке података на само првих неколико главних компоненти како би се добили подаци ниже димензије, уз очување што већег броја варијација података. Прва главна компонента може се еквивалентно дефинисати као правац који максимализује варијанту пројектованих података. Главна компонента $\mathbf{i^{th}}$ може се узети као ортогонални правац првим главним компонентама **i - 1** који максимализује варијанту пројектованих података. На слици 16 се може видети употреба PCA алгоритма.

```
from sklearn.decomposition import PCA
pca = PCA(n_components = 5)
X2 = pca.fit_transform(X2)
explained_variance = pca.explained_variance_ratio_
```

Слика 15 - Употреба PCA алгоритма



Слика 16 - Конфузиона матрица након PCA алгоритма

Закључак

PCA и SVM имају приближну прецизност. SVM даје прецизност од 55.66 процената док PCA даје нешто више, тј. 58.27 процената. Да би се побољшала прецизност, требало би повећати број атрибута, јер на основу само 9 атрибута, не може се добити задовољавајућа прецизност.

Литература

[1] Keel Dataset Description, https://sci2s.ugr.es/keel/dataset.php?cod=58

[2] Python, https://www.python-course.eu/neural_networks_with_python_numpy.php

[3] Machine learning Repository, http://archive.ics.uci.edu/ml/datasets/Statlog+%28Shuttle%29

[4] Pandas documentation, https://pandas.pydata.org/pandas-docs/stable/

[5] SelectKBest, https://www.kaggle.com/jepsds/feature-selection-using-selectkbest?utm_campaign=News&utm_medium=Community&utm_source=DataCamp.com

[6] Chi2, https://machinelearningmastery.com/chi-squared-test-for-machine-learning/

[7] SKlearn documentation, https://scikit-learn.org/stable/documentation.html

[8] PCA, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html