КЛАСИФИКАЦИЈА СКУПА ПЕЧУРАКА НА КЛАСЕ ЈЕСТИВИХ И ОТРОВНИХ

Пројекат у оквиру курса
Истраживање података 1
Математички факултет

Дуња Спасић
mi16073@matf.bg.ac.rs
10. август 2019.

1 УВОД

У овом пројекту је вршена класификација скупа података о 23 врсте печурака на јестиве и отровне. Скуп података је задат са 22 атрибута именског типа који описују станиште, боју, облик делова печурке, мирис, текстуру итд. и 23. атрибутом који даје информацију о класи којој врста печурке припада, тј. да ли је јестива или отровна. Циљ пројекта је да се на основу сакупљених описних података о печуркама применом класификационих метода направе зависности које одређују да ли је печурка јестива или отровна. Подаци се налазе на линку. На слици 1 дат је увид у првих десет од 8124 слогова табеле података. Описни атрибути су скраћени. На слици 2 су дата пуна значења вредности атрибута.

	p	x	s	n	t	p (1)	f	c	n (1)	k	e	e (1)	s (1)	s (2)	w	w (1)	p (2)	w (2)	o	p (3)	k (1)	s (3)	u
1	e	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	g
2	e	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m
3	p	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	u
4	e	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g
5	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g
6	e	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	n	m
7	e	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	m
8	p	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	w	o	p	k	v	g
9	e	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m
10	e	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	g

Слика 1. Приказ првих десет слогова из табеле података

Attribute Information:

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

Слика 2. Информације о атрибутима

За класификацију су коришћени софтвер *SPSS Modeler* и програмски језик *Python* са библиотекама за истраживање података.

2 КОРИШЋЕНЕ МЕТОДЕ

С обзиром да су за опис печурака коришћени искључиво именски подаци, њихова природа не дозвољава поређење два податка (релација ≤), као ни извршавање основних аритметичких операција. Због тога, за калсификацију ових података није могуће применити алгоритме који користе поређење и аритметичке операције како би се донела одлука о класи. За класификацију над оваквим скупом је потребно применити методе које не узимају у обзир дистанцу између елемената скупа, јер се за рачунање растојања користи одузимање, што нема смисла применити над описним вредностима. Стога у решавању овог проблема не може да се примени метод К најближих суседа *(KNN – K Nearest Neighbours)*. Такође, не може да се примени ни Наивна Бајесова метода са претпоставком о Гаусовој расподели, јер не можемо да претпоставимо да вредности описних атрибута имају нормалну расподелу. За решавање задатог проблема коришћене су метода стабла одлучивања, метода случајне шуме и Наивни Бејесов класификатор.

2.1 Стабло одлучивања

Прва метода која је коришћена је класификација помоћу Стабла одлучивања (*Decision Tree Clasifier*). Код ове методе проблем класификације се решава постављањем питања о вредностима атрибута. Низ питања и одговора се представља стаблом одлучивања, где су чворови у графу питања а гране одговори. У листовима стабла су одлуке које је алгоритам донео о класи. Када се добије одговор на постављено питање, уколико не може да се донесе закључак о томе којој класи припадају слогови са тим вредностима атрибута, поставља се ново питање. Ако су атрибути именски (као у случају ових података), може да се направи бинарно стабло или стабло са више грана по слоју у зависности од тога да ли су питања облика “Да ли је вредност атрибута a1 једнака x?” или “Која је вредност арибута a1?”. Оба облика стабла имају своје предности и мане, предност бинарног стабла је што се не повећава у ширину, а предност стабла са више грана је што је његова дубина највише онолика колики је број атрибута табеле. Када се једном направи стабло одлучивања, класа тест слога се лако одлучује једном проласком кроз пут стабла. У *SPSS Modeler*-у је коришћен *CART* (*Classification And Regression Trees*) алгоритам, који прави искључиво бинарна стабла одлучивања.

2.2 Случајна Шума

Друга примењена метода у овом раду је Случајна шума (*Random Decision Forest*). Ова метода се заснива на претходној методи у коју су унете неке измене како би се успешније спречило преприлагођавање скупа за обучавање (тренинг скупа). Алгоритам случајне шуме је ансамбл метода. То су методе које користе неколико алгоритама учења како би побољшали преформансе или тачност предвиђања. У овом случају,

Случајна Шума алгоритам прави неколико стабла одлучивања, а затим као решење враћа моду предвиђених класа. То је класа која је добијена као решење за тај слог у највећем броју покренутих стабала. Број стабала која се користе, у *Python* језику се задаје као аргумент функције класификације. Такође могуће је задати и максималну дубину тих стабала.

2.3 Наивни Бајесов класификатор

Наивни Бајесов класификатор је алгоритам чија је главна идеја примена Бајесове теореме. Бајесова теорема описује везу између вероватноће неког догађаја и претходног знања о условима који су могли да доведу до тог догађаја. Једначина Бајесове теореме гласи:

$$P(Y|X)=\frac{P(X|Y)\cdot P(Y)}{P(X)}$$

$P(Y|X)$ означава вероватноћу да се деси догађај Y онда када се деси догађај X. Ако класе имају недетерминистичку везу са атрибутима онда њихов однос са тим атрибутима можемо да опишемо помоћу условних вероватноћа. Битно је да у тренинг фази алгоритам научи условне вероватноће за сваку комбинацију података добијених из скупа за обучавање. Наивни Бајесов класификатор претпоставља условну независност атрибута за дату класу и тако процењује условну вероватноћу. У зависности од типа и расподеле вредности атрибута, може да се на различите начине примени Наивни Бајесов класификатор. Пошто су подаци именски, не може да се примени Наивни Бајесов класификатор за нормалну расподелу јер су за његову примену потребни непрекидни подаци (нумерички и реални) за које може да се претпостави нормална расподела. Ткође, не може да се претпостави ни Бернулијева расподела, јер за сваки атрибут постоји више од две различите вредности које може да има. Зато је примењен Мултиномијалан Наивни Бајесов класификатор, јер може свакој могућој вредности атрибута да се додели вероватноћа. У *Python* коду су вршене две класификације, са наученом расподелом и са претпостављеном униформном расподелом вредности атрибута.

3 Python ПРОГРАМ

3.1 Фаза препроцесирања

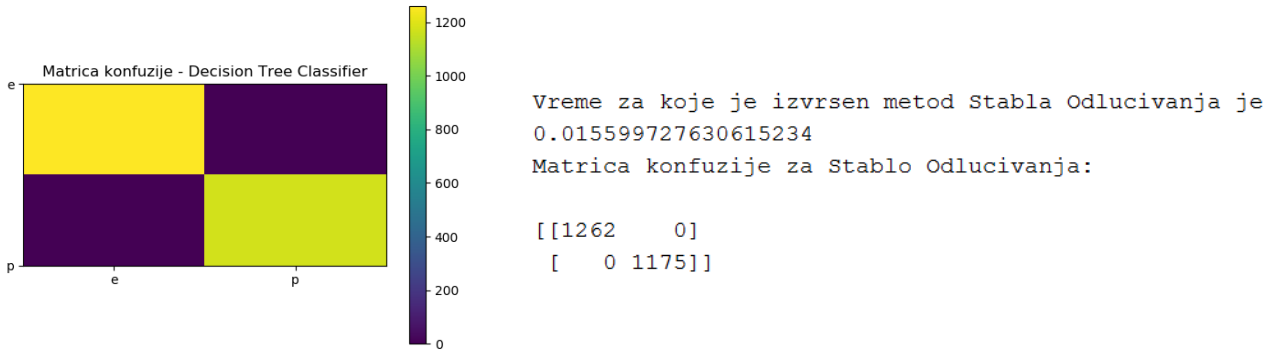
Методе у библиотекама за класификацију података у језику *Python* примају само нумеричке податке. У табели података су сви атрибути именски подаци, који не могу да се пореде, јер означавају станиште, боју печурке и тако даље. Зато је у фази препроцесирања потребно да се подаци кодирају, али тако да вредност њиховог кода не утиче на класификацију. У библиотеци *sklearn.prepocessing* постоји метода *LabelEncoder* која кодира именске податке у нумеричке. Проблем код ове

методе је то што кодира само једнодимезионе низове, тај проблем је решен у коду тако што се у петљи кодира свака колона табеле података појединачно. У фази препроцесирања су још и нормализовани подаци помоћу методе *MinMaxScaler* из библиотеке *sklearn.prepocessing*. Скуп је подељен на две партиције: обучавајући и тест скуп тако да тест скуп чини 30% целог скупа података.

3.2 Класификација

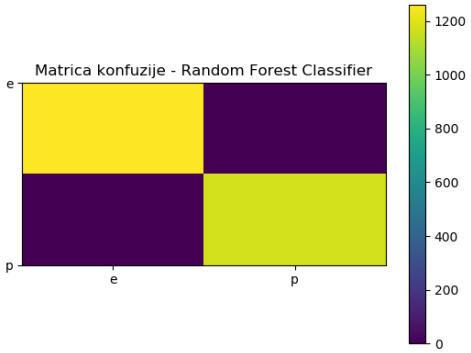
У овом коду су примењени алгоритми Стабло Одлучивања (*DTC – Decision Tree Classifier*), Алгоритам Случајне шуме (RFC – Random Forest Classifier) и Мултиномијални Наивни Бајесов (*Multinomial Naive Bayes*) алгоритам са претпоставкама о униформној расподели (*fit_prior* аргумент је подешен на нетачно) и са наученом расподелом (*fit_prior* аргумент је подешен на тачно). Код доношења одлуке који је најбољи метод треба још узети у обзир и да, осим захтева за што већом тачношћу, треба да буде што мањи проценат лажно јестивих инстанци, тј. што мањи проценат отровних печурака сврстаних у јестиве (у матрицама конфузије претстављено доле лево). Вредности елемената матрице кофузије су различите при сваком покретању програма јер зависе од тога које су инстанце додељене тест, а које обучавајућем скупу. У случају када тест скуп чини 30% почетног скупа, Стабло Одлучивања и Алгоритам Случајне Шуме у највећем броју случаја дају 100% тачну класификацију, а Наивни Бајесов алгоритам даје мањи број лажно јестивих када је *fit_prior* нетачан, тј. када је претпостављена расподела униформна.

Резултати које је програм дао при једном од покретања су приказани на следећим сликама. На слици 3 је дат графички приказ матрице конфузије при покретању алгоритма Стабло Одлучивања, време за које се алгоритам извршио и нумерички приказ матрице конфузије.



Слика 3. Стабло одлучивања: Матрица конфузије, њене вредности и време извршавања алгоритма.

На слици 4 су приказани графички и нумерички приказ резултата алгоритма Случајне шуме и време његовог извршавања.

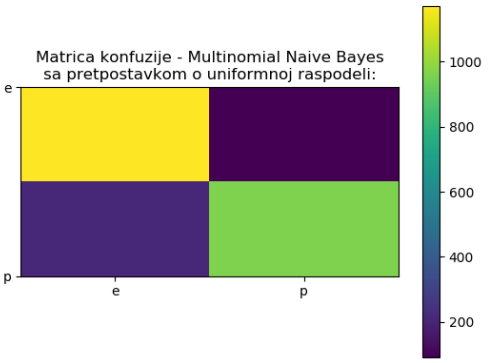



```
Vreme za koje je izvrsen metod Slucajne Sume je
0.015599966049194336
Matrica konfuzije za Slucajnu Sumu:
[[1262    0]
 [   0 1175]]
```

Слика 4. Метод Случајне шуме: Матрица конфузије, њене вредности и време извршавања алгоритма.

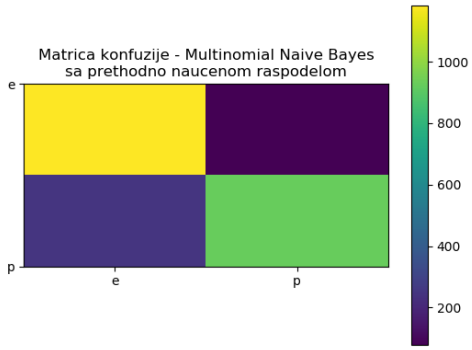
Све инстанце су тачно класификоване алгоритмима Случајне Шуме и Стаблом Одлучивања.

Матрица конфузије Мултиномијалног Наивног Бајесовог алгоритма са претпоставком о униформној расподели, њене нумеричке вредности и време извршавања тог алгоритма дати су на слици 5. Матрица конфузије истог алгоритма са наученом расподелом са временом извршавања тог алгоритма дата је на слици 6.




```
Vreme za koje je izvrsen Multionomijani
Naivni Bajesov algoritam sa unif. raspodelom je
0.0
Matrica konfuzije za Multinomialni Naivni Bajesov
algoritam sa pretpostavkom o uniformnoj raspodeli:
[[1171   91]
 [ 213  962]]
```

Слика 5. Матрица конфузије, њене вредности и време извршавања Мултиномијалног Наивног Бајесовог алгоритма са претпоставком о униформној расподели.




```
Vreme za koje je izvrsen Multionomijani
Naivni Bajesov algoritam sa naucenom raspodelom je
0.015599966049194336
Matrica konfuzije za Multinomialni Naivni Bajesov
algoritam sa prethodno naucenom raspodelom:
[[1184   78]
 [ 249  926]]
```

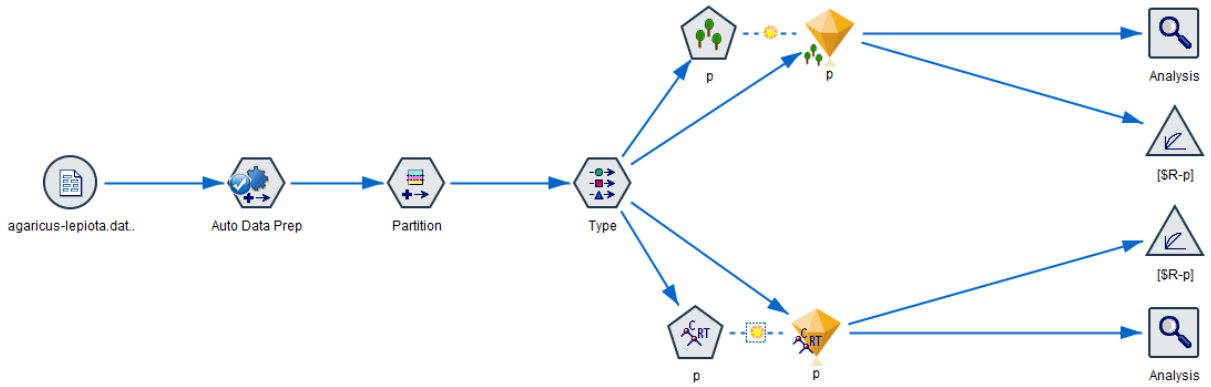
Слика 6. Матрица конфузије, њене вредности и време извршавања Мултиномијалног Наивног Бајесовог алгоритма са претпоставком о претходно наученој расподели.

У обе примене Наивног Бајесовог алгоритма је већи број лажно отровних него лажно јестивих печурака, али са претпоставком о униформној раподели је број лажно јестивих мањи.

Програм мери и пореди и време извршавања свих ових метода, али нема правила која се метода најбрже извршава при различитим покретањима програма. При овом покретању се најбрже извршио Мултиномијални Наивни Бајесов алгоритам са претпоставком о униформној расподели.

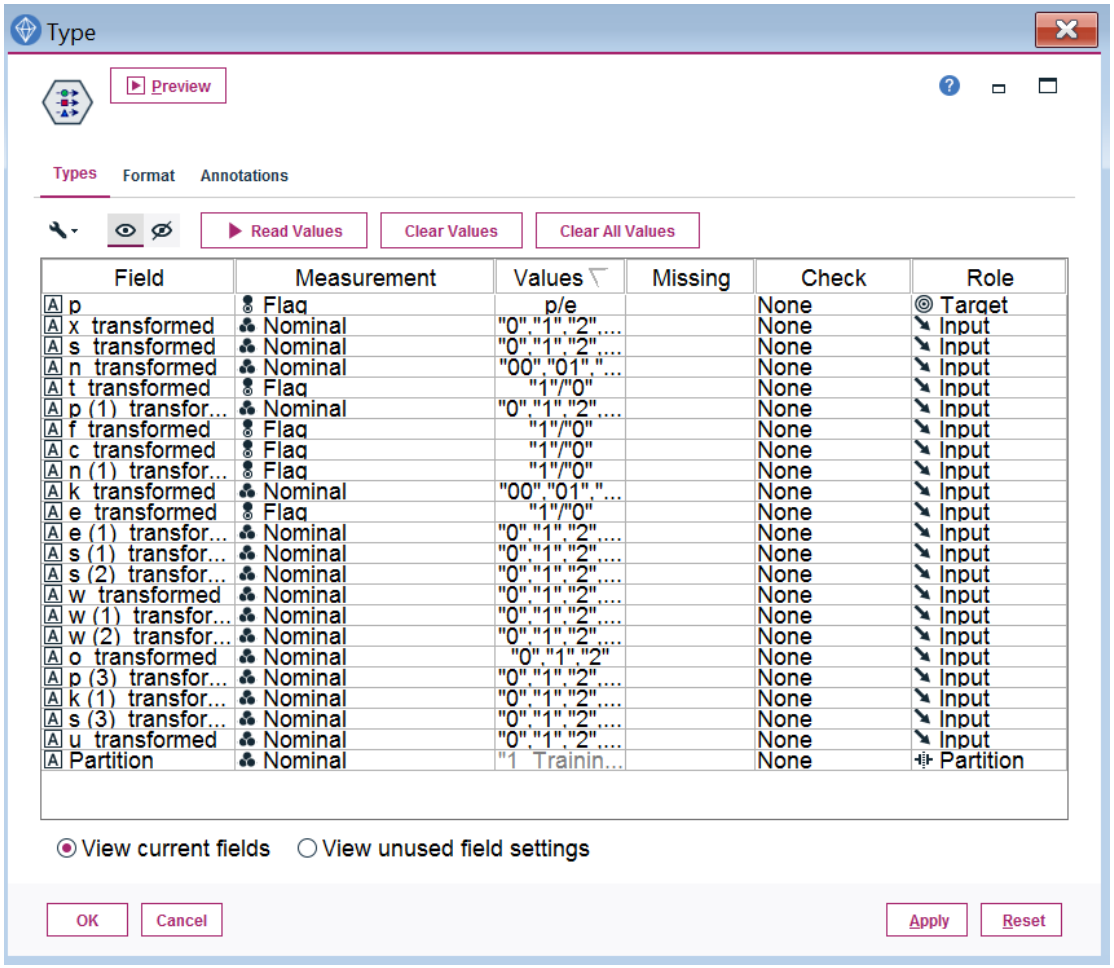
4 РЕШЕЊЕ У *SPSS Modeler*-У

При решавању задатог проблема класификације јестивих и отровних печурака у *SPSS Modeler*-у примењене су методе Случајне шуме и *CART* алгоритам стабла одлучивања. Графички приказ примењене процедуре за класификацију датих података приказан је на слици 7.



Слика 7. Графички приказ примењене процедуре за класификацију података

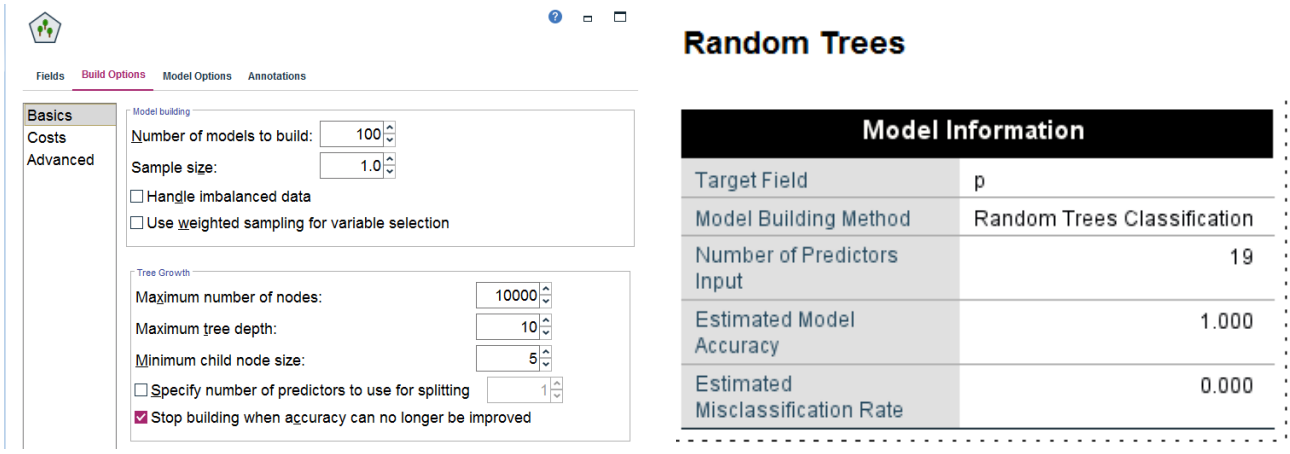
Након учитавања података и припреме, партиционише се скуп података на обучавајући и тест скуп. Скуп за обучавање је подешен да буде 70% почетног скупа, а тест скуп 30%. По учитавању улазних вредности одређен је атрибут који претставља класу и постављен је као циљни атрибут (слика 8).



Слика 8. Особине атрибута табеле

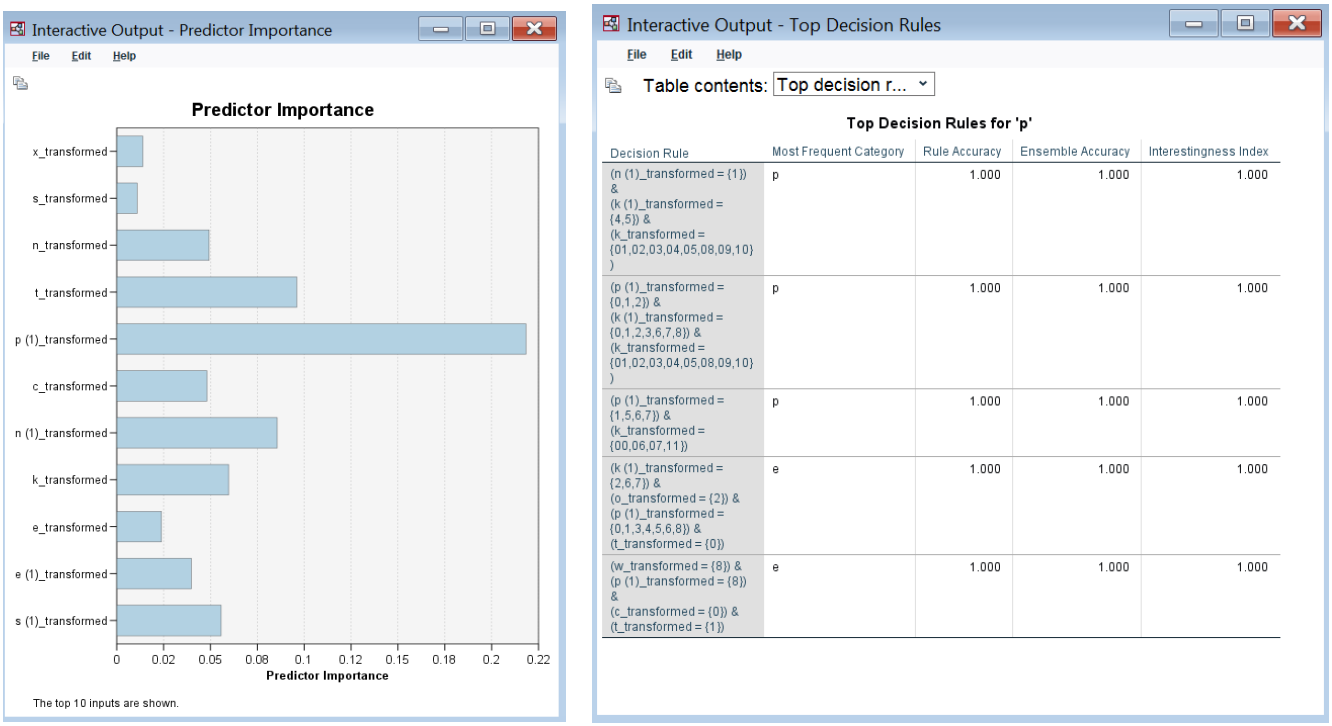
4.1 Примена алгоритма Случајна шума

На сликама 9 и 10 су приказана подешавања за алгоритам Случајне шуме. Формира се 100 стабала за класификацију који имају највише по 10000 чворова од по највише 5 грана са највећом дубином стабла 10. На слици 10. су приказане информације о направљеном моделу случајне шуме.



Слика 9. Подешавања Случаје шуме

Слика 10 (лево) приказује нормирану меру значајности сваког атрибута за одређивање припадности класи, а слика 10. (десно) представља добијена правила за одлучивање, на основу атрибута табеле. На слици 11 је приказана табела са подацима о броју тачно и нетачно класификованих печурака алгоритмом Случајне шуме.



Слика 10. Важност атрибута при класификовању (лево), правила одлучивања (десно).

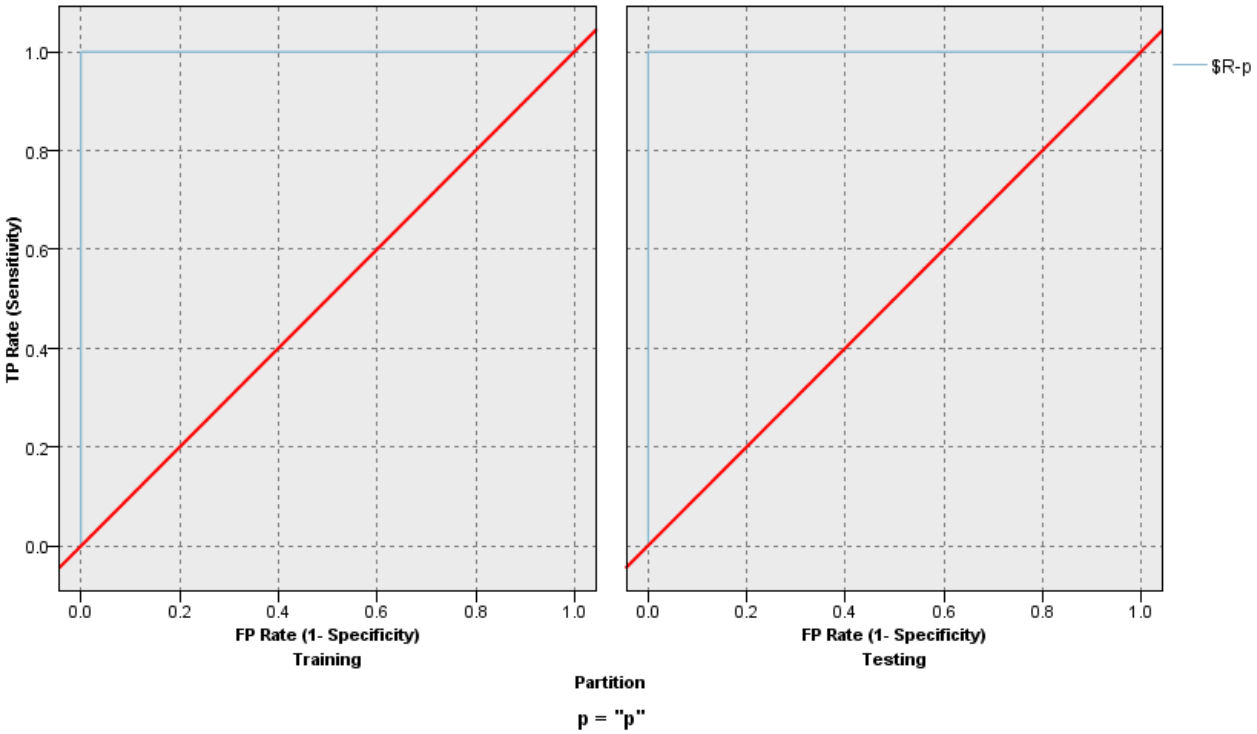
Results for output field p

Comparing $R-p with p

'Partition'	1_Training		2_Testing	
Correct	5,662	100%	2,461	100%
Wrong	0	0%	0	0%
Total	5,662		2,461	

Слика 11. Број и проценат тачно и нетачно класификованих печурака алгоритмом Случајне шуме

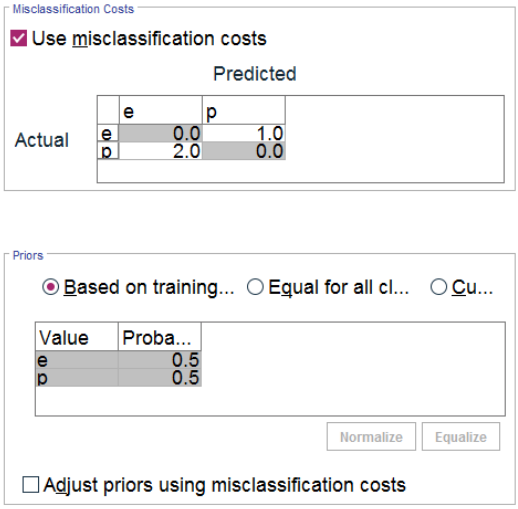
На слици 12. је представљен график *ROC* криве скупа за обучавање и тест скупа, тј. крива која приказује график односа тачно позитивно и лажно позитивно класификованих слогова. Овде се рачуна да је позитивна класа отровних печурака тј. где је вредност атрибута *p* = *“p”*.



Слика 12. График *ROC* криве класе отровних печурака

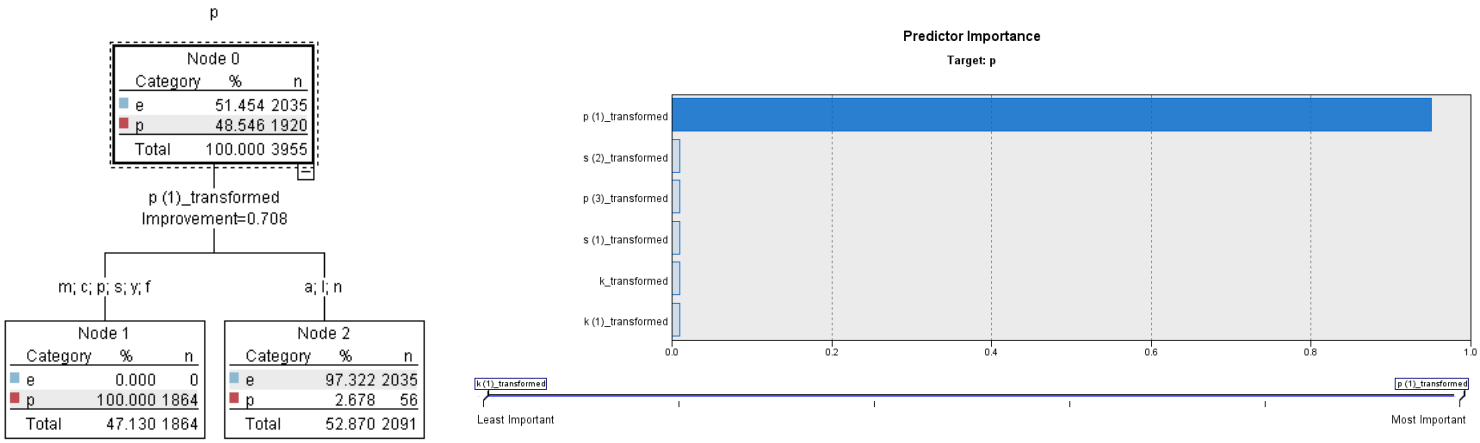
4.2 Примена *CART* алгоритма

На слици 13. су приказана подешавања за алгоритам *CART.* Постављено је да су цене грешке за лажно отровне 1.0 и лажно јестиве 2.0 у жељи да се смањи вероватноћа да се отровне печурке класификују у јестиве. Задато је такође и да су вероватноће припадања слога свакој од класа 0.5.



Слика 13. Подешавања за алгоритам *CART*

На слици 14. (десно) је штапићасти дијаграм који показује значај сваког од атрибута при доношењу одлуке о класи која ће бити додељена слогу. На слици 14. (лево) је бинарно стабло формирано овим алгоритмом. *CART* алгоритам узима само мирис печурке као битну карактеристику у одлучивању да ли је отровна јер су све печурке са мирисима *creosote, fishy, foul, musty, pungent, spicy* отровне у 100% случајева. Има 56 оних које су отровне али не миришу тако (приказано на слици 14 лево). Ово је за алгоритам мала грешка али не сме да се дозволи да се отровне печурке класификују у јестиве, па 2 није довољно повећање цене. Цена грешке за лажно јестиву печурку је повећавана до 33. До 32,9 алгоритам прави идентично стабло оном које је приказано на слици 14 десно, док се са ценом грешке 33 све печурке класификују у отровне.



Слика 14. Стабло одлучивања (лево), дијаграм значајности атрибута при одлучивању (десно)

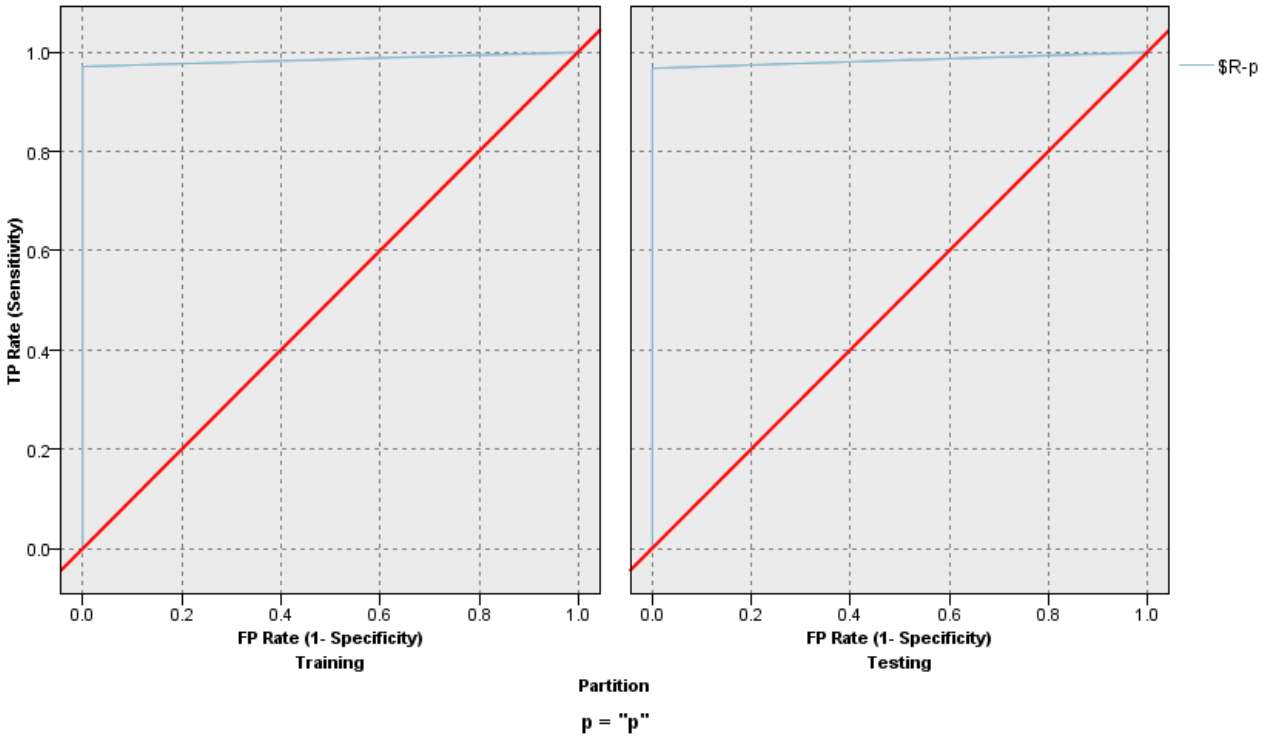
У табели на слици 15. је приказан проценат тачно и нетачно класификованих података у обучавајућем скупу и скупу за тестирање. График *ROC* криве *CART* алгоритма је приказан на слици 16.

Results for output field p

Comparing $R-p with p

'Partition'	1_Training		2_Testing	
Correct	5,580	98.55%	2,423	98.46%
Wrong	82	1.45%	38	1.54%
Total	5,662		2,461	

Слика 15. Број и проценат тачно и нетачно класификованих печурака *CART* алгоритмом



Слика 16. График *ROC* криве *CART* алгоритма

5 ЗАКЉУЧАК

У *Python* решењу проблема су се много боље показали алгоритми Стабла одлучивања (укључујући и Случајну шуму) од Наивног Бајесовог алгоритма. Наивни Бајесов алгоритам није дао потпуно тачну класификацију ни у једном покретању програма. Када се смањи величина скупа за обучавање, Стабло одлучивања и Случајна шума дају различите тачности, мада без неког одређеног правила тако да је немогуће донети закључак који је тачнији. При подели где је 70% почетног скупа скуп за обучавање, а 30% тест скуп, оба алгоритма дају потпуно тачну класификацију (100% тачности).

Осим тога што Наивни Бајесов метод није ни једном дао потпуно тачну класификацију, већи је број отровних печурака које класификује као јестиве, него број јестивих које класификује као отровне. Ово је грешка коју је требало потпуно минимизовати, а идеално је да се 0 отровних печурака класификује као јестиво. Иако не простоји разлика у проценту тачности када се за Мултиномијалан Наивни Бајесов алгоритам користи научена расподела и униформна расподела, коришћењем униформне расподеле углавном се добија мањи број лажно јестивих печурака.

Није примећено правило који се од четири алгоритама примењених над овим подацима најбрже извршава. Главни критеријум у доношењу одлуке о томе који је алгоритам најбољи за примену је постигнута тачност. Као што је приказано матрицама конфузије ове четири методе, Случајна шума и Стабло одлучивања дају идеалну тачност. Са оваквом процентуалном поделом на обучавајући и тест скуп, увек дају потпуно

тачну класификацију. Зато може да се закључи да су погодни за коришћење над овим скупом података о печуркама.

У *SPSS Modeler*-у су примењени *CART* алгоритам и алгоритам Случајне шуме са 100 стабала. Овако задана Случајна шума даје тачну класификацију за све елементе скупа података. Ова тачност је битна пре свега јер не класификује ни једну отровну печурку као јестиву. С друге стране, *CART* алгоритам прави јако мало стабло, поставља само једно питање на основу кога класификује печурке у јестиве и отровне. Пресудни атрибут је мирис печурке. Све печурке које имају мирисе *creosote, fishy, foul, musty, pungent, spicy* су отровне. Проблем је што алгоритам све остале печурке сврстава у јестиве, јер је мали проценат оних које су отровне а не миришу овако. Иако је цена повећана дупло за класификацију отровних као јесиве, премала је грешка да би алгоритам поставио и једно друго питање. Када се цена грешке повећа до 32,9 *CART* алгоритам прави потпуно исто стабло као и када су грешке једнаке. Када је грешка погрешне класификације отровних у јестиве 33, алгоритам све печурке сврстава у отровне. Прави се мало стабло, што је велика предност, али ипак овај алгоритам не може да се користи јер прави грешку која је превише озбиљна, иако је мала. Стога се може закључити да је Алгоритам Случајне шуме ипак бољи приступ за класификацију овог скупа података.