Klasifikacija pečuraka

Seminarski rad u okviru kursa
Istraživanje podataka
Matematički fakultet

Tatjana Radovanović

1. oktobar 2019

Sažetak

Svrha ovog rada je da ispita da li se pečurke mogu razvrstati na jestive i otrovne samo na osnovu izgleda. Prilikom rešavanja ovog problema upotrebljene su različite metode klasifikacije. Za obradu podataka korišćen je programski jezik Python.

Sadržaj

1 Uvod

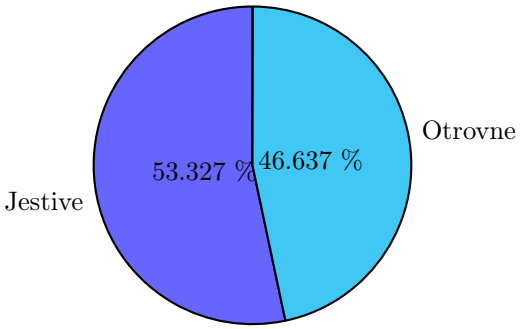
Pečurke se od davnina koriste u ljudskoj ishrani. Kao što je poznato, neke su jestive, a neke veoma otrovne, čak i smrtonosne. Ljudi su vremenom zapazili da otrovne pečurke imaju neke karakteristike koje jestive nemaju i obrnuto. Cilj ovog rada je da utvrdi koliko je ova metoda pouzdana, tj. da li sa sigurnošću možemo tvrditi da je neka pečurka jestiva samo na osnovu njenog opisa bez daljih hemijskih analiza.

Poglavlje 2 upoznaje čitaoce sa podacima i objašnjava upotrebljene tehnike pretprocesiranja. U poglavlju 3 prikazane su različite metode klasifikacije i njihovi rezultati. Metode koje su obrađene su drveta odlučivanja, k najbližih suseda, naivni Bajesov klasifikator, veštačke neuronske mreže i metod potpornih vektora.

Za obradu podataka korišćene su biblioteke programskog jezika Python. Modeli konstruisani za potrebe ovog rada se mogu naći na ovoj adresi.

2 Podaci

Podaci su preuzeti sa linka. Korišćen je prošireni skup podataka koji sadrži 8416 redova sa 23 atributa. Svi atributi su kategorički i odnose se na fizičke karakteristike pečuraka. U skupu se nalazi 53.327% jestivih i 46.637% otrovnih pečuraka. Odnos jestivih i otrovnih pečuraka je prikazan na grafiku 2.



Slika 1: Odnos jestivih i otrovnih pečuraka

2.1 Pretprocesiranje podataka

Pre pravljenja modela potrebno je pripremiti podatke. Vrednosti atributa 'Edible', koji uzima vrednosti EDIBLE i POISONOUS, 'ručno' su postavljeni na 1 i 0 kako bi se osiguralo da jedinica označava da je pečurka jestiva. Slično radimo i sa atributom 'Bruises' gde vrednost BRUISES menjamo sa 1, a vrednost NO sa 0.

Nedostajuće vrednosti u našem skupu podataka su označene '?'. Vrednost '?' menjamo sa nan iz numpy biblioteke. Daljom analizom se utvrđuje da su sve nedostajuće vrednosti vezane za atribut 'Stalk-root' i da čine približno 29,47% svih vrednosti tog atributa. Zbog toga je odlučeno da se taj atribut ukloni iz našeg skupa.

Uklonjen je i atribut 'Veil-type'. Razlog za takvu odluku je taj što ovaj atribut ima samo jednu vrednost na celom skupu.

Kako su svi atributi kategorički potrebno je izvršiti dodatno pretprocesiranje. Za to je upotrebljeno binarno kodiranje.

```
df = pd.get_dummies(df)
```

Listing 1: Binarno kodiranje

Dodatno, vršena je i standardizacija podataka. Za to je korišćen StandardScaler za sve metode klasifikacije osim za naivni Bajesov klasifikator. StandardScaler transformiše vrednosti atributa tako da srednja vrednost bude 0, a standardna devijacija 1. Standardna ocena za uzorak x se računa po sledećoj formuli

$$z = \frac{x - u}{s} \tag{1}$$

gde je u srednja vrednost, a s standardna devijacija[1]. Kod naivnog Bajesovog klasifikatora je korišćen MinMaxScaler jer je potrebno da svi atributi budu pozitivni. MinMaxScaler transformiše atribute tako što ih skalira u zadatom opsegu. Ako opseg nije zadat, kao što je slučaj u našem modelu, onda se koristi podrazumevani rang (0, 1)[1].

3 Klasifikacija

Zadatak ovog rada je da napravi model koji vrši klasifikaciju pečuraka u jestive i otrovne. Metode koje su upotrebljene kako bi se rešio ovaj problem su:

- drvo odlučivanja
- k najbližih suseda
- naivni Bajesov klasifikator
- veštačke neuronske mreže
- metod potpornih vektora

Za ciljnu promenljivu y uzet je atribut 'Edible', a x su ostali atributi. Pre obučavanja svakog od ovih metoda potrebno je izvršiti podelu na podatke za trening i podatke za testiranje. Za testiranje je uzeto 30% podataka i izvršena je stratifikacija po ciljnoj promenljivoj.

```
x_train, x_test, y_train, y_test = train_test_split(x, y,
    test_size = 0.3, stratify = y)
```

Listing 2: Podela na podatke za trening i obučavanje

Bitno je napomenuti da je standardizacija podataka izvršena nakon podele na trening i test skup. Razlog tome je taj što se podaci koji se koriste za evaluaciju modela ni na koji način ne smeju koristiti prilikom njegovog obučavanja. Ukoliko bi se standardizacija primenila na ceo skup podataka pre podele na trening i test skup tada bi vrednosti atributa iz skupa za testiranje uticali na prosek i standardnu devijaciju koji se koriste pri standardizaciji i samim tim imali uticaj i na skup za običavanje[2].

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

Listing 3: Standardizacija skupova za obučavanje i testiranje

Prilikom pravljenja modela za k najbližih suseda, drvo odlučivanja, neuronske mreže i metod potpornih vektora korišćena je unakrsna validacija. Ovom tehnikom se lakše pronalaze parametri koji daju najbolje rezultate.

3.1 Drvo odlučivanja

Za unakrsnu validaciju koristimo drveta odlučivanja sa različitim parametrima. Kao kriterijumi podele se koriste entropija i Ginijev kriterijum, maksimalna dubina stabla je između 3 i 10[1], a maksimalan broj listova je između 2 i 10.

```
parameters = [{'criterion' : ['gini', 'entropy'],
               'max_depth' : range(3, 10),
               'max_leaf_nodes' : range(2, 10),
}]

dt = GridSearchCV(DecisionTreeClassifier(), parameters, cv = 5)

dt.fit(x_train, y_train)
```

Listing 4: Pravljenje drveta odlučivanja

Najbolji rezultati se dobijaju kada se za kriterijum podele koristi entropija i kada je maksimalna dubina 4, a maksimalan broj listova 8. Uspeh takvog klasifikatora se procenjuje na 0.999 $\pm$ 0.001 Na skupu za testiranje preciznost datog klasifikatora za klasu 0 iznosi približno 1, a za klasu 1 približno 0.997. Odziv za klasu 0 je približno 0.9966, a za klasu 1 približno 1. Matrica konfuzije je data u tabeli 1.

Tabela 1: Matrica konfuzije za drvo odlučivanja

	0	1
0	1174	4
1	0	1347

3.2 K najbližih suseda

Svi modeli k najbližih suseda koji su analizirani unakrsnom validacijom daju dobre rezultate na skupu za validaciju. Uspeh svakog klasifikatora je oko 0.999. Broj suseda je u rasponu od 1 do 9. Kao parametri rastojanja Minkovskog se uzimaju 1 (Menhentn rastojanje) i 2 (Euklidsko rastojanje). Težina suseda može biti 'uniform' i 'distance'[2].

```
parameters = [{'n_neighbors': range(1,9),
               'p':[1, 2],
               'weights': ['uniform', 'distance'],
               }]

knn = GridSearchCV(KNeighborsClassifier(), parameters, cv=5)

knn.fit(x_train, y_train)
```

Listing 5: Obučavanje modela k najbližih suseda

[1]Ne uključujući 10. Npr. range(1, 4) su brojevi 1, 2 i 3

[2]Kod 'uniform' svi susedi imaju jednak uticaj dok kod 'distance' veći uticaj imaju bliži susedi

Klasifikator koji koristi Menhentn rastojanje i čija je težina suseda uniformna, a broj suseda jednak 1 ima stopostotan uspeh na validacionom skupu. Na test skupu ponovo ima idealne rezultate, i preciznost i odziv za obe klase iznosi 1. Matrica konfuzije je prikazana u tabeli 2

Tabela 2: Matrica konfuzije za k najbližih suseda

	0	1
0	1178	0
1	0	1347

3.3 Naivni Bajesov klasifikator

Kod Naivnog Bajesovog klasifikatora nije upotrebljena unakrsna validacija i koriste se podrazumevani parametri. Ovaj klasifikator je precizno klasifikovao 2444 instance od 2525 koliko ih se nalazi u test skupu. Preciznost za klasu 0 iznosi približno 0.9937, a za klasu 1 približno 0.9477. Odziv za klasu 0 je približno 0.9372, a za klasu 1 približno 0.9948. U tabeli 3 je prikazana matrica konfuzije.

```
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

mnb = MultinomialNB()
mnb.fit(x_train, y_train)
```

Listing 6: Obučavanje Naivnog Bajesovog klasifikatora

Tabela 3: Matrica konfuzije za Naivni Bajesov klasifikator

	0	1
0	1104	74
1	7	1340

3.4 Veštačke neuronske mreže

Za obučavanje neuronkih mreža je korišćen rešavač za optimizaciju težina stohističkog opadajućeg gradijenta. Maksimalan broj iteracija je 500. Konstruisane su mreže sa različitim parametrima. Stopa učenja pri ažuriranju težina (learning_rate) može biti konstantna (constant), može se postepeno smanjivati u koraku t (invscaling) i može da se ne menja dok se vrednost funkcije gubitka smanjuje (adaptive). Inicijalna stopa učenja (learning_rate_init) je 0.01, 0.005, 0.002 i 0.001. Kao aktivacione funkcije (activation) se koriste identička funkcija (identity)[3], sigmoidna funkcija (logistic)[4], tanges hiperbolički (tanh)[5] i ispravljena linearna jedinica (relu)[6]. Broj neurona u skrivenim slojevima je (10, 3) ili (10, 10).

[3] f(x) = x
[4] $\sigma(x) = \frac{1}{1+e^{-x}}$
[5] $tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$
[6] relu(x) = max(0, x)

```
params = [{'solver':['sgd'],
           'learning_rate':['constant', 'invscaling', 'adaptive'],
           'learning_rate_init':[0.01, 0.005, 0.002, 0.001],
            'activation' : ['identity', 'logistic', 'tanh', 'relu'
    ],
            'hidden_layer_sizes' : [(10,3), (10,10),],
           'max_iter': [500]

           }]

mlp = GridSearchCV(MLPClassifier(), params, cv=5)
mlp.fit(x_train, y_train)
```

Listing 7: Obučavanje neuronskih mreža

Uspeh različitih neuronskih mreža na validacionom skupu je vrlo raznolik. Najgori klasifikator ima uspeh $0.480\,, \pm 0.053\,$, a uspeh najboljeg iznosi $1.0\ \pm 0.1$ Kod najboljeg modela aktivaciona funkcija je identička, broj neurona u skrivenim slojevima je (10, 3), stopa učenja pri ažuriranju težina je 'adaptive' i inicijalna stopa učenja je 0.01. Na test skupu ova neuronska mreža za klasu 0 ima preciznost 1, a za klasu 1 približno 0.9993. Odziv za klasu 0 je približno 0.9992, a za klasu 1 iznosi 1. Matrica konfuzije je data u tabeli 4.

Tabela 4: Matrica konfuzije za neuronsku mrežu

	0	1
0	1177	1
1	0	1347

3.5 Metod potpornih vektora

Parametri koji su korišćeni za unakrsnu validaciju su sledeći: kao parametar regularizacije korišćene su vrednosti 0.01, 0.1 i 1, a kao funkcije kernela se zadaju 'linear', 'poly' i 'sigmoid'. Uspeh ovih klasifikatora na validacionom skupu je raznolik. Najlošiji klasifikator ima uspeh $0.695\,\mathrm{d} \pm 0.024\,$dok najbolji klasifikatori imaju stopostotni uspeh.

```
parameters = [{'C': [0.01, 0.1, 1],

               'kernel' : ['linear', 'poly', 'sigmoid']

}]


svm = GridSearchCV(SVC(), parameters, cv=5)

svm.fit(x_train, y_train)
```

Listing 8: Unakrsna validacija kod metoda potpornih vektora

Kao najbolji model izdvaja se onaj kod koga je parametar regularizacije 0.1, a funkcija kernela 'linear'. Preciznost i odziv na test skupu iznose 1 za obe klase. Matrica konfuzije se nalazi u tabeli 5

4 Zaključak

Svi upotrebljeni metodi klasifikacije daju zadovoljavajuće rezultate. Najbolje se pokazao metod k najbližih suseda koji je za sve parametre u

Tabela 5: Matrica konfuzije za potporne vektore

	0	1
0	1178	0
1	0	1347

unakrsnoj validaciji imao uspeh preko 99%, a klasifikator koji se pokazao najboljim imao je idealno predviđanje na test skupu. Za njim sledi metod potpornih vektora čiji najbolji klasifikator takođe tačno razvrstava pečurke iz test skupa. Ipak, ovaj metod ne daje tako dobre rezultate za sve parametre korišćene prilikom unakrsne validacije i zbog toga se metod k najbližih suseda smatra boljim.

Najbolji klasifikator kod neuronske mreže pravi jako malu grešku na test skupu. Drvo odlučivanja takođe daje dobre rezultate, ali malo lošije nego neuronska mreža. Ipak, trebalo bi napomenuti da neuronska mreža ne radi dobro za sve ispitane parametre, čak za neke daje loše rezultate, dok kod drveta odlučivanja nisu primećene tolike oscilacije uspeha za različite parametre.

Naivni Bajesov klasifikator se najlošije pokazao u ovom zadatku. Iako daje dobre rezultate greši dosta više nego drugi klasifikatori. Trebalo bi uzeti u obzir činjenicu da je ovom metodu posvećeno najmanje pažnje i da su za njegovo obučavanje upotrebljeni podrazumevani parametri, kao i da nije korišćena unakrsna validacija.

Rezultati ovog rada pokazuju da je moguće na osnovu fizičkih karakteristika pečurake utvrditi da li je jestiva ili otrovna. Ipak, sa njihovom konzumacijom treba biti oprezan jer u prirodi ništa nije 100% sigurno i pečurka koja nije opisana u ovom skupu podataka može iznenaditi i najbolji klasifikator.

Literatura

[1] scikit-learn. on-line at: https://scikit-learn.org/stable/.

[2] Mladen Nikolić i Anđelka Zečević. Mašinsko učenje, 2019.