

MUSHROOM CLASSIFICATION

INGEGNERIA DELLA CONOSCENZA

A.A. 2018-2019

Realizzato da:

Aleksandar Kostić 662585 (kostic.aleksandar97@gmail.com)

Amedeo Racanati 663528 (a.racanati4@studenti.uniba.it)

INDICE

Introduzione	3
Riconoscimento dei funghi velenosi	4
Descrizione dei dati	4
Funzionamento	7
Riconoscimento dei sintomi	9
Knowledge base	9
Funzionamento	11
Specifiche tecniche	13
Casi d'uso	14

Introduzione

Questo sistema software è stato ideato al fine di aiutare quelle persone che desiderano riconoscere le varie specie di funghi velenose o commestibili.

Ci sono svariate caratteristiche che determinano la commestibilità di un fungo.

Pertanto, quando l'utente vuole conoscerne la commestibilità, tramite l'utilizzo di questo sistema gli vengono poste delle domande, al fine di ottenere tutte quelle informazioni utili alla classificazione del fungo.

Inoltre, qualora una persona abbia ingerito un fungo ed inizi ad avvertire dei particolari sintomi, può interrogare il sistema affinché possa sapere se abbia ingerito un fungo velenoso o meno.

Il sistema si articola in due sezioni separate. Una sezione si occupa di determinare la commestibilità di un fungo, date le sue caratteristiche; mentre l'altra sezione permette di scoprire se una persona abbia ingerito un fungo velenoso o meno, dati i suoi sintomi.

Riconoscimento dei funghi velenosi

Descrizione dei dati

Il sistema utilizza dei classificatori che vengono istruiti da dei dati di training affinché possano classificare nuovi funghi. Qui ci soffermeremo sui dati. In seguito verrà spiegato il funzionamento dei classificatori.

Il dataset di training è stato prelevato dal repository online di UCI Machine Learning. Esso include la descrizione dei funghi di 23 specie diverse, in lingua inglese. Ogni specie è classificata come commestibile o velenosa. Per quei funghi dove non si può stabilire con certezza la commestibilità, la loro classificazione rientra tra i funghi velenosi.

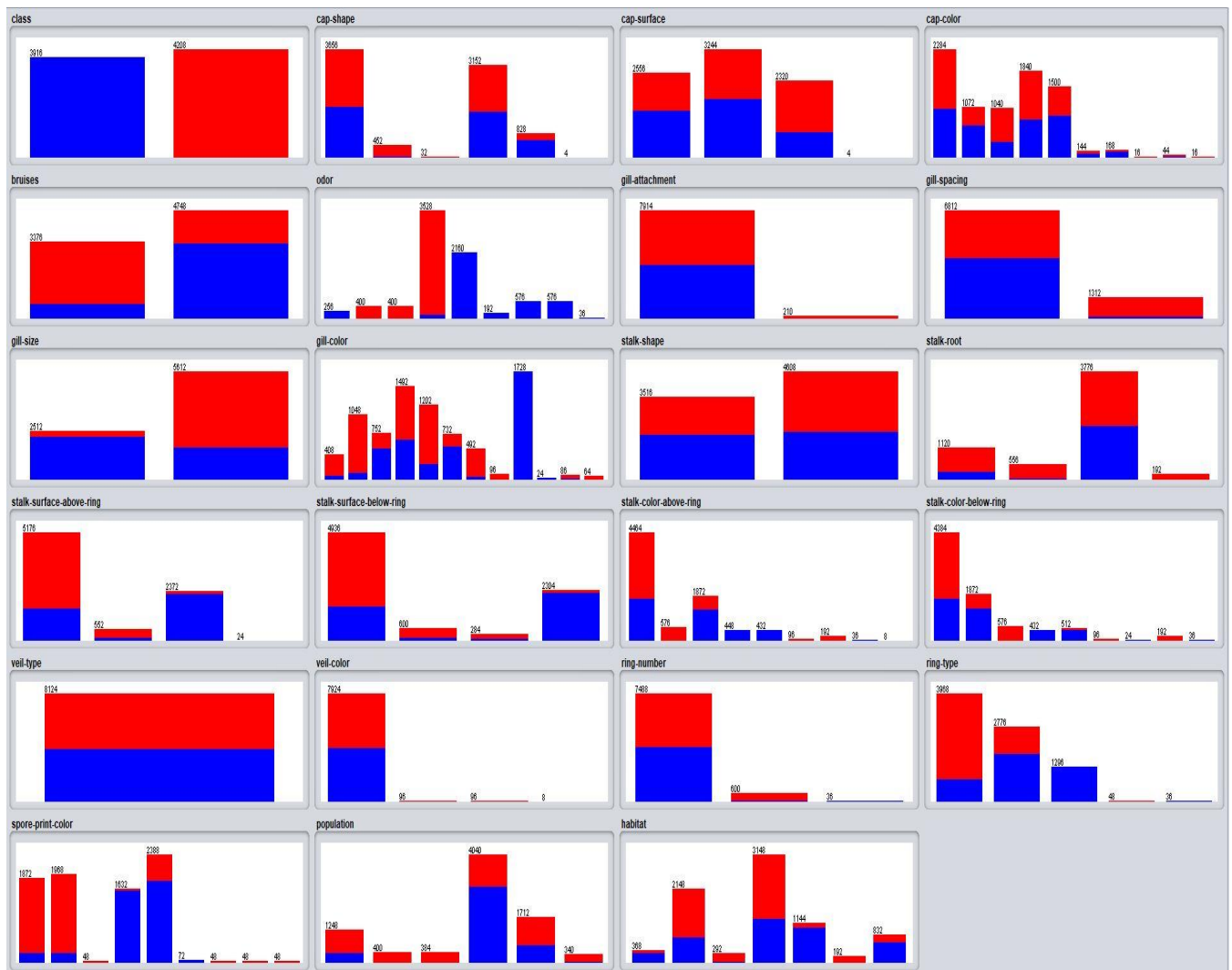
Il dataset è un file in formato csv, dove ogni riga corrisponde ad uno specifico fungo, classificato opportunamente. Qui di seguito vengono elencati gli attributi di ciascuna riga con i possibili valori:

- Class: edible=e, poisonus=p;
- Cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s;
- Cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s;
- Cap-color: brown=n, buff=b, cinnamon=c, gray= g, green=r, pink=p, purple=u, red=e, white=w, yellow=y;
- Bruises: bruises=t, no=f;
- Odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s;
- Gill-attachment: attached=a, descending=d, free=f, notched=n;

- Gill-spacing: close=c, crowded=w, distant=d;
- Gill-size: broad=b, narrow=n;
- 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;
- Stalk-shape: enlarging=e, tapering=t;
- Stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?;
- Stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s;
- Stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s;
- Stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y;
- Stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y;
- Veil-type: partial=p, universal=u;
- Veil-color: brown=n, orange=o, white=w, yellow=y;
- Ring-number: none=n, one=o, two=t;
- Ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z;
- Spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y;
- Population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y;

- Habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d.

La distribuzione delle classi di appartenenza per ciascun attributo è mostrata qui di seguito:



Funzionamento

Il sistema, al fine di poter effettuare classificazioni di nuovi funghi, deve necessariamente disporre di un dataset di training, dal quale poter addestrare i suoi modelli di classificazione. Durante la classificazione vengono utilizzati due modelli differenti: un albero di decisione e una rete bayesiana.

Ciascun classificatore possiede dei pro e dei contro. L'albero di decisione funziona bene quando il dataset di training non possiede errori nei dati e quando tutte le feature che concorrono alla determinazione della classe sono esplicitate, ossia non sono presenti variabili latenti. La rete bayesiana invece riesce a trattare meglio i valori mancanti, rispetto all'albero di decisione, e riesce anche a rilevare eventuali variabili latenti, ma qualora queste condizioni vengano meno, è in genere meno preciso rispetto ad un albero di decisione.

Una volta che l'utente ha specificato tutte le informazioni a disposizione relative al fungo, il sistema procede con la classificazione. I due classificatori vengono utilizzati in maniera combinata, al fine di trarre il massimo vantaggio da entrambi.

Innanzitutto si controlla che l'utente abbia specificato, riguardo al fungo, le informazioni necessarie per l'albero di decisione.

In caso affermativo, si procede con la classificazione tramite albero di decisione. Qualora questa classificazione non si ritenga soddisfacente, viene combinata la classificazione dell'albero con quella della rete bayesiana.

In caso negativo (ossia quando il fungo non possiede determinate informazioni importanti per l'albero di decisione), viene utilizzata soltanto la classificazione tramite rete bayesiana, che permette di trattare più efficacemente i valori mancanti.

L'albero di decisione utilizza l'algoritmo C4.5 per la creazione del modello, ed in più effettua una potatura dei rami, qualora ciò fosse necessario.

La rete bayesiana invece crea una belief network in maniera dinamica, e stima le probabilità direttamente dai dati di training, non potendo disporre di altra conoscenza a priori.

Il sistema inoltre permette di visualizzare le informazioni principali dei due classificatori, corredate dei risultati inerenti la valutazione degli stessi, valutazione effettuata tramite una 10-folds validation. Infine permette anche la possibilità di effettuare nuovi addestramenti dato un nuovo dataset di training.

Riconoscimento dei sintomi

Knowledge base

Lo scopo di questa sezione del sistema è quella di determinare la sintomatologia di una persona, qualora egli accusi dei sintomi che possono essere legati all'ingerimento di un fungo velenoso.

I rapporti di causa ed effetto legati ai sintomi sono stati modellati tramite l'utilizzo delle clausole definite proposizionali, che in seguito verranno meglio esplicitate.

Queste clausole formano la knowledge base del sistema, che viene utilizzata per inferire la sintomatologia dell'utente e capire se abbia ingerito un fungo velenoso.

La conoscenza necessaria alla definizione della knowledge base è stata raccolta dai vari articoli presi dal web. La base di conoscenza qui di seguito espressa è stata definita dinamicamente all'interno del sistema, e può subire delle variazioni qualora si voglia raffinare la conoscenza espressa.

```
poisonous_mushroom <- long_incubation ^ long_symp_occourence
```

```
poisonous_mushroom <- short_incubation ^ short_symp_occourence
```

```
long_incubation <- synd_phallotoxin
```

```
long_incubation <- synd_gyromitrin
```

synd_phallotoxin <-symp_gastroenteritis ^ symp_abdominal_cramping

synd_phallotoxin <-symp_diarrhea ^ symp_nausea ^

symp_increased_hepatic_enzymes

synd_phallotoxin <-symp_hepatic_failure ^ symp_renal_failure

synd_phallotoxin <-encephalopathy

synd_gyromitrin <-symp_gastroenteritis ^ symp_dizziness ^

symp_headache ^ symp_intractable_seizures

synd_gyromitrin <-oliguria

synd_gyromitrin <-symp_acute_renal_failure

synd_gyromitrin <-symp_tubuloinerstitial_nephritis ^ symp_fibrosis

short_incubation <- synd_muscarine

short_incubation <- synd_coprine

short_incubation <- synd_psilocybin

short_incubation <- synd_gi_toxins

synd_coprine <-symp_facial_flushing ^ drink_alcohol

synd_coprine <-symp_nausea ^ drink_alcohol

synd_coprine <-symp_vomiting ^ drink_alcohol

synd_coprine <-symp_hypotension ^ drink_alcohol

synd_psilocybin <-symp_euphoria ^ symp_hallucinations

synd_psilocybin <-symp_agitation ^ symp_hyperthermia

```
synd_gi_toxins <-symp_diarrhea ^ ~drink_alcohol  
synd_gi_toxins <-symp_vomiting ^ ~drink_alcohol  
synd_gi_toxins <- symp_nausea ^ symp_diarrhea
```

Funzionamento

Tramite l'utilizzo delle clausole definite proposizionali, ossia della knowledge base definita precedentemente, può essere inferita la tossicità del fungo ingerito.

Quando un utente vuole sapere se il fungo ingerito sia velenoso, la query iniziale posta al sistema è la seguente:

```
yes <- poisonous_mushroom
```

Una volta che l'utente effettua la domanda al sistema, viene effettuato un procedimento di risoluzione Top-down, il quale inizia con la risoluzione della query precedente.

Siccome il sistema non può conoscere in anticipo i sintomi della persona, provvede a chiederglieli durante la fase risolutiva della query. Infatti gli

vengono poste diverse domande, le quali sono in seguito utilizzate per dedurre lo stato della persona.

Il sistema inoltre adopera l'assunzione di conoscenza completa, la quale permette di inferire un numero maggiore di clausole.

Una volta che è stata inferita la velenosità del fungo, l'utente ha la possibilità di visualizzare l'albero risolutivo della query da lui posta, ossia di comprendere quali sono le clausole utilizzate per l'inferenza, e quindi per la determinazione della risposta. Ciò permette di aumentare la fiducia che l'utente ha verso la risposta data dal software.

Specifiche tecniche

Il sistema è stato implementato in Java, e l'interfaccia grafica è di tipo console. I classificatori usati fanno parte della libreria Weka per Java, opportunamente configurati; mentre il sistema che effettua la deduzione delle clausole proposizionali è stato implementato nel programma stesso.

Qualora si voglia classificare un nuovo fungo, è necessario che i classificatori siano stati allenati almeno una volta. Infatti senza dei dati di training non sarebbe possibile trarre alcuna sorta di conclusione.

La knowledge base invece è definita nello stesso codice Java. Pertanto una modifica della knowledge base può essere effettuata solo tramite una modifica del programma.

Casi d'uso

Qui di seguito vengono mostrate varie schermate per i diversi scenari d'uso definiti precedentemente.

```
Welcome to the Mushroom knowledge program.

Select an option:
1. Enter to the mushroom classification
2. You are not feeling well? Find out if you have eaten a poisonous mushroom
3. Exit

Enter choice:1
Select an option:
1. Use training set
2. Display Decision Tree
3. Display Bayesian Network
4. Classify new instance
5. Exit

Enter choice:1
Please specify a csv file, in order to train the system to recognize poisonous and edibles mushrooms (specify -1 in order to exit):
C:\Users\kosti\eclipse-workspace\Shrooms\mushrooms.csv
Select an option:
1. Use training set
2. Display Decision Tree
3. Display Bayesian Network
4. Classify new instance
5. Exit
```

Effettuato un training dei classificatori, specificando il file csv dei dati

```

== MODEL ==
J48 pruned tree
-----
odor = p: p (256.0)
odor = a: e (400.0)
odor = l: e (400.0)
odor = n
|   spore-print-color = k: e (1296.0)
|   spore-print-color = n: e (1344.0)
|   spore-print-color = u: e (0.0)
|   spore-print-color = h: e (48.0)
|   spore-print-color = w
|   |   gill-size = n
|   |   |   gill-spacing = c: p (32.0)
|   |   |   gill-spacing = w
|   |   |   |   population = s: e (0.0)
|   |   |   |   population = n: e (0.0)
|   |   |   |   population = a: e (0.0)
|   |   |   |   population = v: e (48.0)
|   |   |   |   population = y: e (0.0)
|   |   |   |   population = c: p (16.0)
|   |   |   gill-size = b: e (528.0)
|   |   spore-print-color = r: p (72.0)
|   |   spore-print-color = o: e (48.0)
|   |   spore-print-color = y: e (48.0)
|   |   spore-print-color = b: e (48.0)
odor = f: p (2160.0)
odor = c: p (192.0)
odor = y: p (576.0)
odor = s: p (576.0)
odor = m: p (36.0)

Number of Leaves   :    24
Size of the tree   :    29

```

Descrizione dell'albero di decisione

```

== EVALUATION ==

Correctly Classified Instances      8124           100    %
Incorrectly Classified Instances      0             0    %
Kappa statistic                      1
Mean absolute error                  0
Root mean squared error              0
Relative absolute error              0    %
Root relative squared error          0    %
Total Number of Instances           8124

Select an option:
1. Use training set
2. Display Decision Tree
3. Display Bayesian Network
4. Classify new instance
5. Exit

Enter choice:

```

Descrizione della precisione dell'albero di decisione

```

Select a property to insert:
1. cap shape
2. cap surface
3. cap color
4. bruises
5. odor
6. gill attachment
7. gill spacing
8. gill size
9. gill color
10. stalk shape
11. stalk root
12. stalk surface above ring
13. stalk surface below ring
14. stalk color above ring
15. stalk color below ring
16. veil type
17. veil color
18. ring number
19. ring type
20. spore print color
21. population
22. habitat

-1. Begin classification

-1
Mushroom: cap shape=bell; cap color=green; odor=foul; stalk surface below ring=smooth; population=abundant; habitat=paths
Your mushroom is edible, with probability of 100.00 %

```

Esempio di classificazione di un fungo, una volta specificate le sue caratteristiche

```

Select an option:
1. Enter to the mushroom classification
2. You are not feeling well? Find out if you have eaten a poisonous mushroom
3. Exit

Enter choice:2
Did you eat the mushroom 6 hours ago or earlier? (y/n)
n
Did you eat the mushroom later than 6 hours ago? (y/n)
y
Did you accuse symp facial flushing? (y/n)
y
Did you drink alcohol in the past hours? (y/n)
y
Your mushroom is poisonous. Do you want to know why? (y/n)

```

Esempio di domande e risposte atte a riconoscere la tossicità di un fungo. Esito positivo


```

Select an option:
1. Enter to the mushroom classification
2. You are not feeling well? Find out if you have eaten a poisonous mushroom
3. Exit

Enter choice:2
Did you eat the mushroom 6 hours ago or earlier? (y/n)
y
Did you accuse symp gastroenteritis? (y/n)
n
Did you accuse symp diarrhea? (y/n)
n
Did you accuse symp hepatic failure? (y/n)
n
Did you accuse symp encephalopathy? (y/n)
n
Did you accuse symp oliguria? (y/n)
n
Did you accuse symp acute renal failure? (y/n)
n
Did you accuse symp tubulointerstitial nephritis? (y/n)
n
Did you eat the mushroom later than 6 hours ago? (y/n)
n
Your mushroom is not poisonous, take a rest.

```

Esempio di domande e risposte atte a riconoscere la tossicità di un fungo. Esito negativo

```

Select an option:
1. Enter to the mushroom classification
2. You are not feeling well? Find out if you have eaten a poisonous mushroom
3. Exit

Enter choice:2
Did you eat the mushroom 6 hours ago or earlier? (y/n)
n
Did you eat the mushroom later than 6 hours ago? (y/n)
y
Did you accuse symp facial flushing? (y/n)
y
Did you drink alcohol in the past hours? (y/n)
y
Your mushroom is poisonous. Do you want to know why? (y/n)
y
yes <-- poisonous_mushroom

Press -1 to exit or a number X for the explanation of the atom at the position X
0
poisonous_mushroom <-- short_symp_occurrence and short_incubation

Press -1 to go up or a number X for the explanation of the atom at the position X
0
short_symp_occurrence <-- true

Press -1 to go up

```

In questo esempio viene mostrato l'albero risolutivo della query, che ha avuto un esito positivo