CLASSIFICAZIONE SU CAMPIONI DI FUNGHI

PROGETTO MACHINE LEARNING

2019/2020

LANTIERI MIRKO - 858278

REMON MIGUEL - 861466

ZANABONI MANUEL - 816105

INTRODUZIONE

Per il progetto di Machine Learning abbiamo scelto come soluzione di partenza il dataset dal titolo **Mushroom** fornito dal sito *UCI Machine Learning Repository* (disponibile qui). I dati sono stati ricavati dalla guida *"The Audubon Society Field Guide to North American Mushrooms"*.

Questo dataset contiene la descrizione di ipotetici campioni corrispondenti a 23 specie di funghi della famiglia Agaricus e Lepiota. La variabile target è il risultato di una combinazione delle 3 classificazioni presenti nella guida, ovvero "decisamente commestibile, decisamente velenoso e commestibilità sconosciuta", queste ultime due classificazioni sono state unite in modo da rendere binaria la classificazione. Questa ipotesi ha senso in quanto, dato il contesto, è decisamente preferibile un falso negativo (ovvero, considerando la classe "commestibile" come positiva, il fungo in questione è dichiarato velenoso ma in realtà è commestibile), rispetto ad un falso positivo. Utilizzeremo questo ragionamento anche per la valutazione dei modelli predittivi presentati in seguito.

Abbiamo scelto questo dataset per la sua quantità di istanze e attributi, le quali ci hanno permesso di lavorare con una buona quantità d'informazioni. Inoltre la copertura quasi completa dei valori (pochi valori mancanti) ci ha permesso di operare su dati di buona qualità.

DESCRIZIONE DEL DATASET

Il dataset utilizzato ha le seguenti caratteristiche:

- Numero di istanze: **8124**
- Caratteristica del dataset: **multivariato**
- Caratteristica degli attributi: **categorici**
- Numero di attributi (variabile target esclusa): **22**
- Valori mancanti (missing): **si**

Dominio degli attributi:

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	bruisesyes=t, bruisesno=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, rhizo-morphs=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

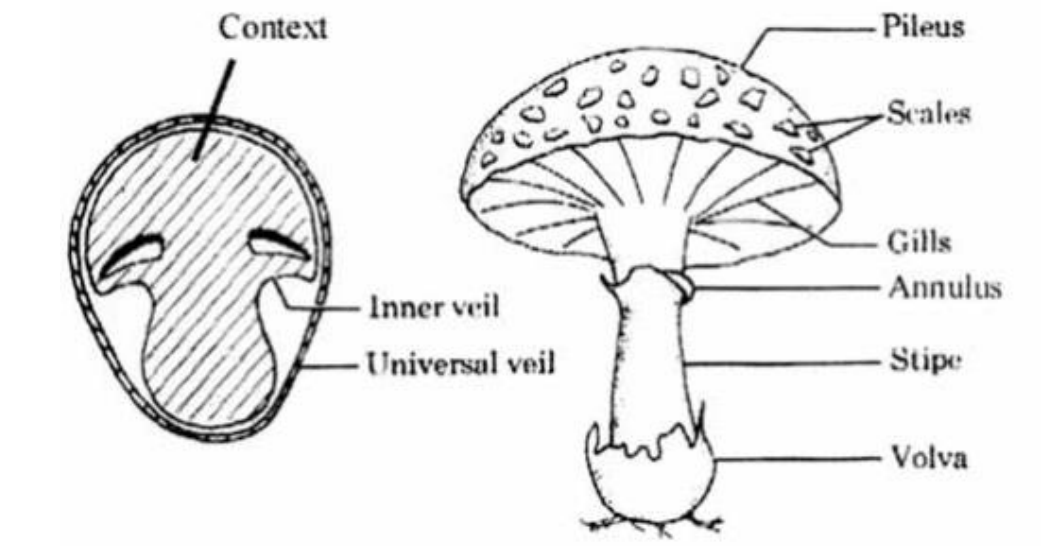
stalk-color-above-ring	brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
stalk-color-below-ring	brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
veil-type	partial=p, universal=u
veil-color	brown=n, orange=o, white=w, yellow=y
ring-number	none=n, 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

Tabella 1: Lista degli attributi

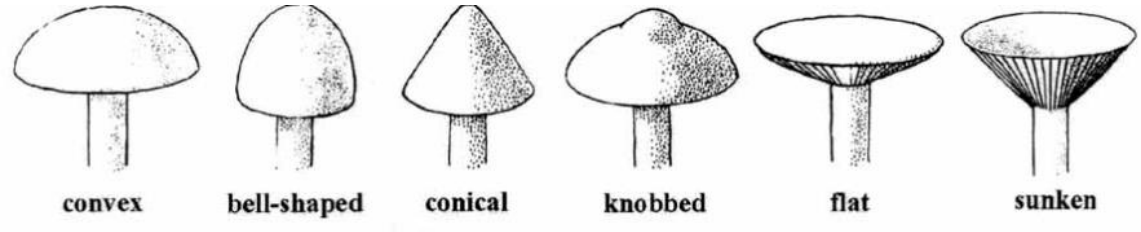
Il dataset è fornito in formato *csv* (comma separated values) e i valori sono memorizzati tramite singoli caratteri. Il mapping tra significato semantico e corrispettivo carattere è presentato nella Tabella 1. Notiamo inoltre che è presente solamente un attributo con possibili valori mancanti, tale attributo è *stalk.root* (valore ?).

Per conoscere le varie caratteristiche dei funghi, vengono mostrate in seguito alcune immagini che le rappresentano, secondo gli attributi presenti nel dataset.

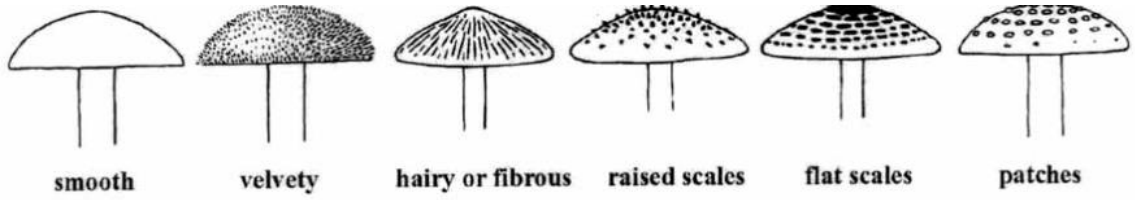
- Mushroom structure:



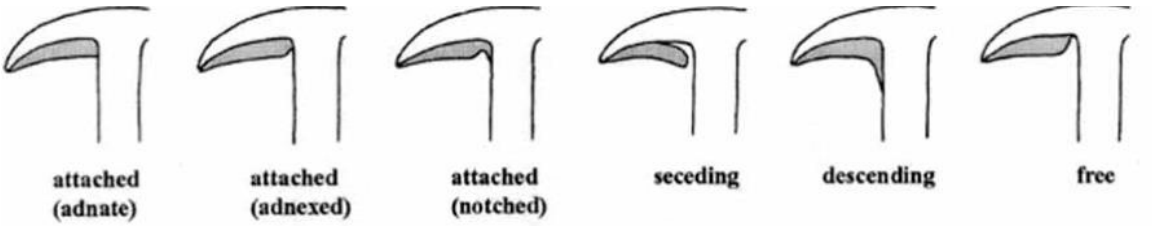
- Mushroom cap shape:



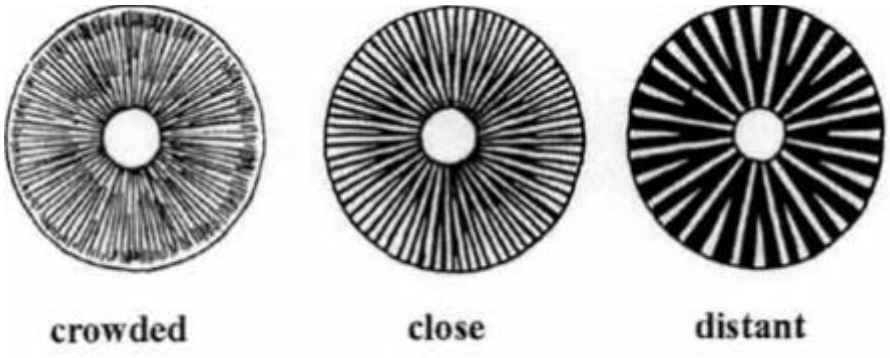
- Mushroom cap surface:



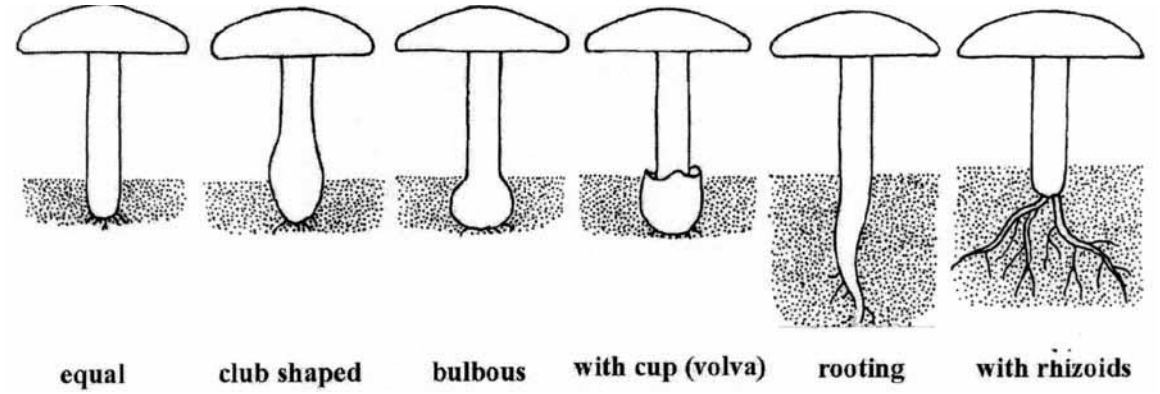
- Mushroom gill attachment:



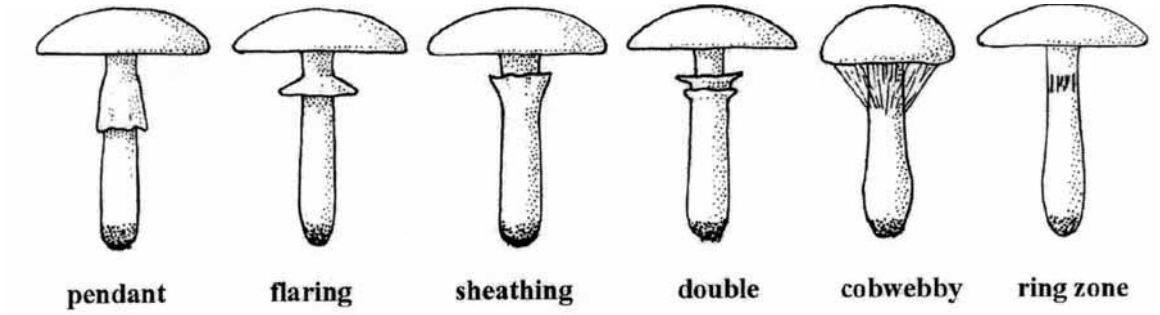
- Mushroom gill attachment:



- Mushroom stalk type:



- Mushroom stalk type:



OBIETTIVI DELL'ELABORATO

L’obiettivo stabilito per questo progetto è lo sviluppo di due modelli di apprendimento che abbiano lo scopo di classificare le istanze secondo le classi “commestibile” e “velenoso”. I modelli di riferimento comprendono l’apprendimento attraverso l’algoritmo degli **alberi decisionali** e le **macchine a vettori di supporto.** Questa scelta deriva dal fatto che:

1. Gli alberi decisionali si adattano meglio alla classificazione di istanze descritte da attributi categorici rispetto ad altri algoritmi di apprendimento.
2. Le macchine a vettori di supporto, anch’esse applicabili ad attributi categorici, performano una classificazione binaria sulle istanze del dataset, denotando un iperpiano di separazione.

I passi sono svolti in base alla seguente procedura: inizialmente abbiamo importato il dataset in formato tabellare csv disponibile dal database *UCI Machine Learning Repository* come menzionato sopra. Dopo di che abbiamo svolto un'analisi esplorativa da cui è possibile ricavare informazioni importanti e misurazioni necessarie per poter determinare l’andamento dei modelli di apprendimento. In seguito si svolge la fase di test di ogni modello, performando in tal modo le misurazioni delle performance su ciascuno di esso, ottenendo risultati sia grafici che tabellari.

ANALISI ESPLORATIVA ED ELABORAZIONE

In seguito al caricamento dei dati abbiamo condotto un’analisi esplorativa del dataset per coglierne possibili informazioni nascoste a prima vista. Sulla base di queste osservazioni verranno modellati gli algoritmi di apprendimento con lo scopo aumentare qualità quali performance, accuratezza, generalizzazione.

La prima manipolazione del dataset è stata quella di rendere i dati leggibili, sostituendo singoli caratteri del dataset con le parole secondo il mapping fornito dal sito.

Visualizzando i livelli di ogni attributo, notiamo che l’attributo *veil.type* presenta solamente il valore “partial”. Ciò significa che la varianza di tale attributo è nulla quindi non aumenta il quantitativo di informazione del dataset nel processo di apprendimento ma ne aumenta solamente la complessità computazionale. Perciò possiamo ignorare questo attributo.

Gestione dei valori mancanti

Per quanto riguarda i valori mancanti (N/A) abbiamo plottato la classificazione delle istanze in base all'attributo *stalk.root* e abbiamo notato che tale covariata non sembra separare bene le istanze (non è correlata alla variabile target). L'unico valore che presenta una classificazione omogenea è "rooted" tuttavia presenta una numerosità di 192, ovvero solamente il 2,4% delle istanze del dataset. L'attributo è stato rimosso.

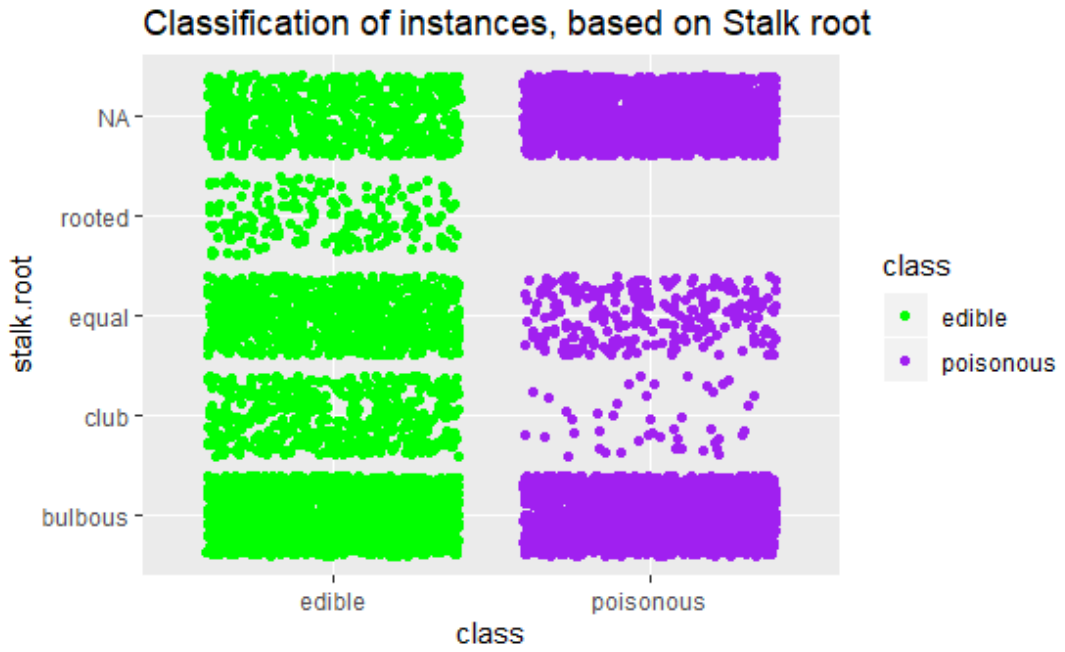


Immagine 1: Classificazione delle istanze basata su stalk.root.

Visualizzazione delle distribuzioni delle istanze in base agli attributi:

Siamo passati successivamente a rappresentare le relazioni tra vari attributi in modo da comprendere se fossero correlati alla variabile target (*class*). Per la visualizzazione abbiamo utilizzato la funzione *ggplot()* sfruttando la geometria jitter.

Per il primo plot (immagine 2) abbiamo messo in relazione gli attributi "cap.color" e "bruises" e abbiamo notato che, seppur non sembrino avere alta correlazione con la variabile target, le combinazioni di valori "red-bruiseyes" e "yellow-bruisesyes" denotano solamente istanze commestibili mentre le combinazioni "red-bruisesno" e "yellow-bruisesno" denotano solamente istanze velenose. Inoltre non sono presenti istanze con combinazioni "green-bruisesyes" e "purple-bruisesyes".

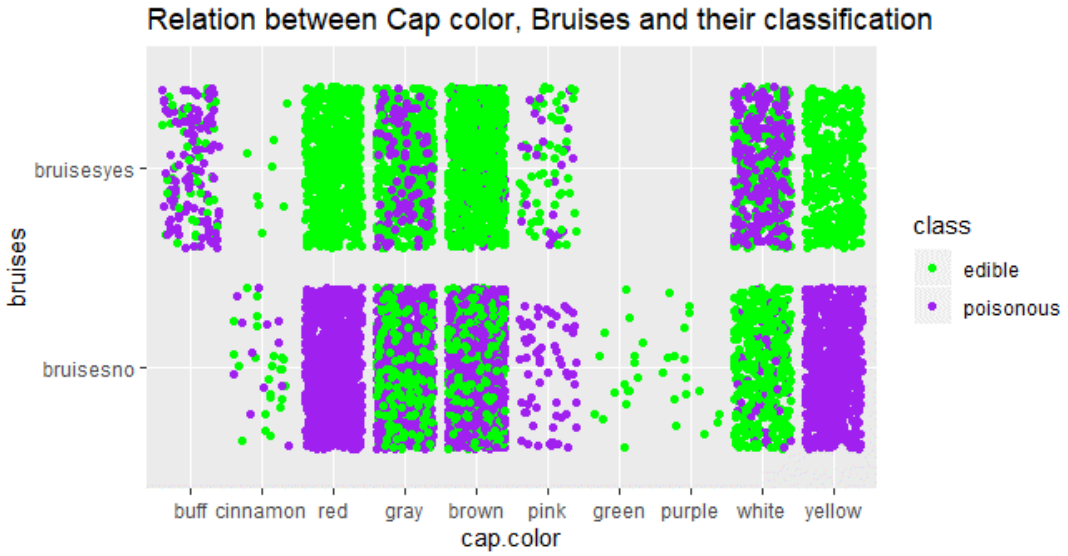


Immagine 2: Relazione cap.color – bruises.

Nella visualizzazione seguente viene rappresentata la relazione tra gli attributi *gill.color* e *spore.print.color* con le rispettive classificazioni. Notiamo che tutte le istanze che presentano la combinazione "buff-white" sono classificate come velenose, in particolare la rappresentazione è molto densa, simbolo di buona numerosità. Inoltre le istanze sembrano generalmente ben divise (buona parte delle combinazioni di valori presenta una classificazione uniforme). Queste considerazioni ci fanno pensare che gli attributi in questione possano essere abbastanza informativi per quanto riguarda la classificazione.

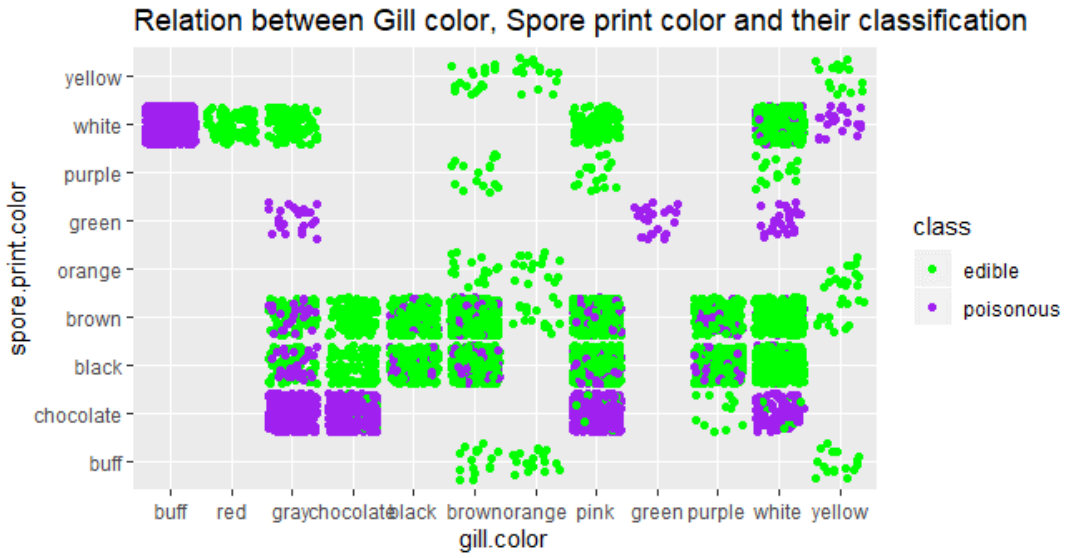


Immagine 3: Relazione gill.color – spore.print.color.

La relazione tra le covariate *odor* e *population* mostra una caratteristica interessante del dataset, sembrano essere molto informative rispetto alla classificazione delle istanze. Infatti la maggior parte delle combinazioni presentano una distribuzione uniforme di classificazioni. In particolare notiamo che, leggendo per righe il plot, l'attributo *odor* si allinea quasi perfettamente alla classificazione della variabile target.

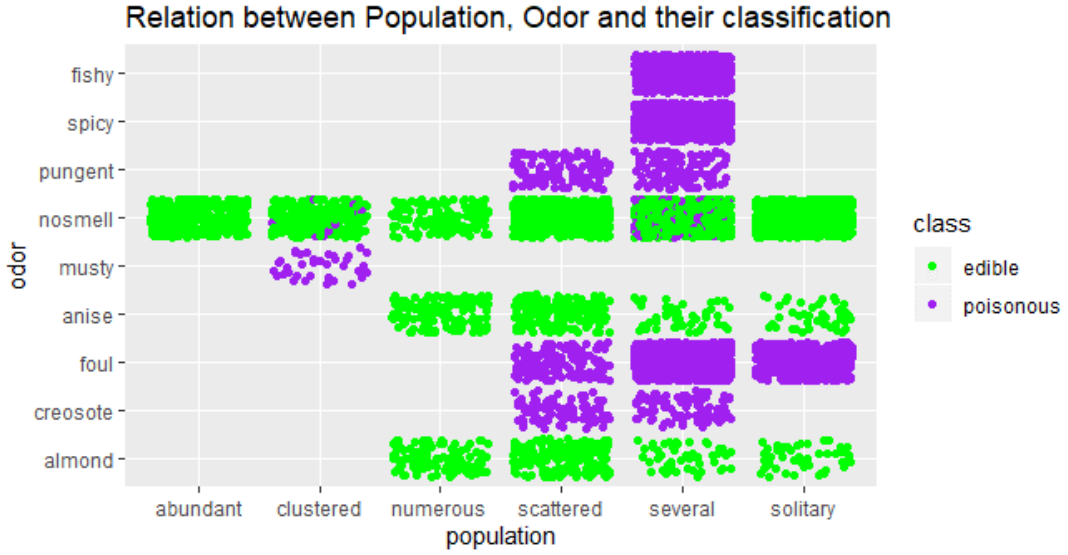


Immagine 4: Relazione population – odor.

Abbiamo quindi intuito che l'attributo *odor* è piuttosto significativo quindi andiamo a visualizzare le classificazioni delle istanze in base ai valori che può assumere.

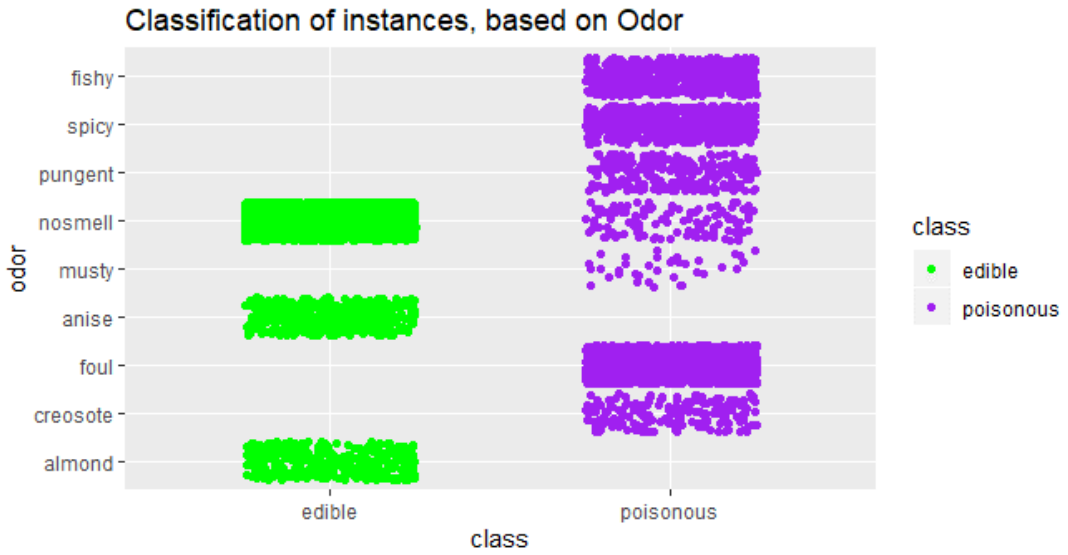


Immagine 5: Classificazione delle istanze in base a odor.

Dal plot precedente notiamo che tutti i valori dell'attributo *odor* tranne "nosmell" classificano perfettamente le istanze. Siamo quindi di fronte a un attributo molto allineato alla variabile target (forse anche troppo). Siamo andati ad analizzare quindi come si comportano quelle istanze con valore dell'attributo *odor* uguale a "nosmell" in relazione ad altre covariate e abbiamo notato una divisione quasi perfetta nell'attributo *spore.print.color*. L'immagine 6 mostra questo andamento.

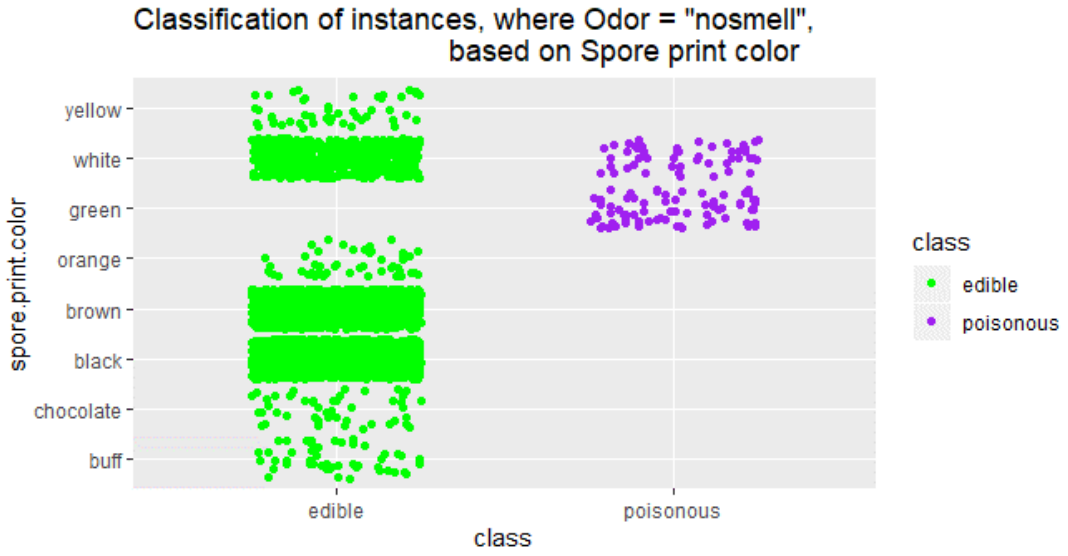


Immagine 6: Classificazione delle istanze aventi attributo odor = "nosmell", basata sull'attributo spore.print.color.

Una visualizzazione dei medesimi dati ma in forma tabellare:

		edible	poisonous
spore.print.color	**buff**	48	0
	chocolate	48	0
	black	1296	0
	brown	1344	0
	orange	48	0
	green	0	72
	purple	0	0
	white	576	48
	yellow	48	0

Notiamo che l'unico valore che presenta classificazioni miste è "white", tuttavia la numerosità dell'etichetta "commestibile" (576) è di gran lunga maggiore dell'etichetta "velenoso" (48).

Date le precedenti osservazioni ipotizziamo che le covariate *odor* e *spore.print.color* siano significative a tal punto da essere determinanti per la classificazione delle istanze.

Concludiamo l'analisi esplorativa andando a visualizzare la numerosità di istanze classificate come commestibili e velenose tramite l'utilizzo di un barplot. Possiamo notare che le classi sono bilanciate.

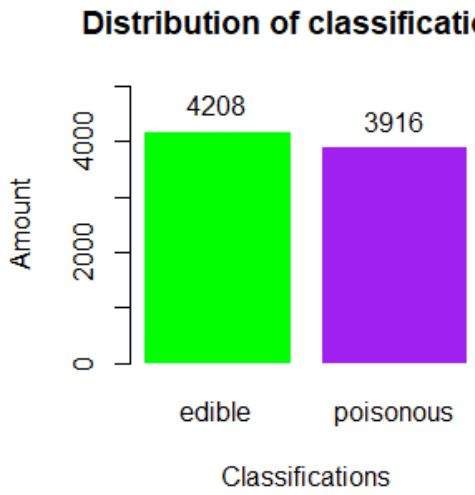


Immagine 7: Rapporto tra istanze commestibili e velenose.

Partizionamento del dataset

In questa fase di sviluppo effettuiamo il partizionamento del dataset, dividendolo in un sottoinsieme dedicato al training dei modelli (riservato all'incirca il 70% del volume del dataset) e la parte rimanente, ossia il 30% del volume dei dati, viene dedicato al testing dei modelli di apprendimento. L'esecuzione di tali modelli sul test set restituisce una predizione sulla classificazione delle istanze. La funzione *split.data* è stata utilizzata per ottenere tale partizionamento (il parametro "seed" rende possibile un partizionamenti randomico di volta in volta) . In seguito al partizionamento otteniamo un training set di 5686 istanze e un test set di 2437 istanze.

SVILUPPO DEI MODELLI DI APPRENDIMENTO

Primo modello: Albero Decisionale

Come da titolo, il primo modello utilizzato è quello dell'albero di decisione (**rpart**).

Inizialmente abbiamo eseguito l'allenamento su tutti gli attributi iniziali del training set, senza specificare un parametro di complessità custom, con lo scopo di verificare se le ipotesi fatte in precedenza vengano confermate. All'eseguire della predizione e della matrice di confusione, il risultato che ne fuoriesce aiuta a comprendere che occorre un possibile "overfitting" iniziale data la classificazione quasi ideale (99.55% di accuratezza). Inoltre notiamo che le variabili considerate sono *odor* e *spore.print.color* come previsto nelle ipotesi.

Allenando il modello con le sole variabili di ipotesi (*odor* e *spore.print.color*) otteniamo lo stesso risultato del modello precedente. A tal punto è comprensibile il fatto che tali variabili dominano il classificatore, risultando così in un modello affetto da overfitting. Nell'immagine seguente presentiamo l'albero ottenuto.

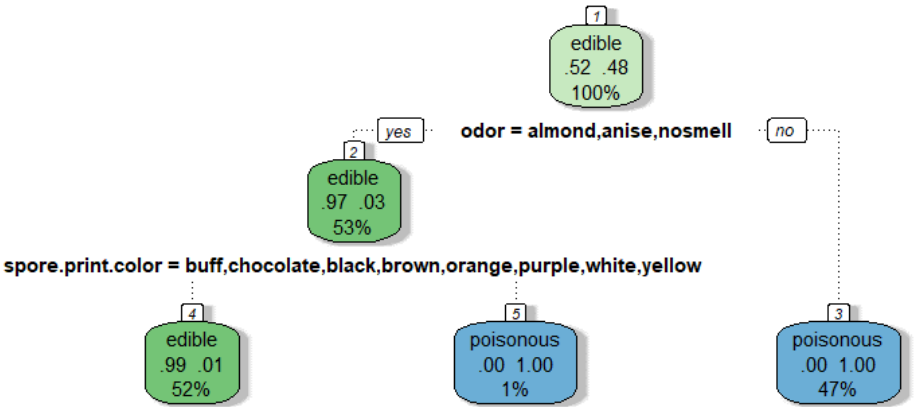


Immagine 8: Albero decisionale iniziale (2 splits).

L’idea per risolvere il problema dell’overfitting è quella di allenare un albero decisionale senza considerare gli attributi *odor* e *spore.print.color* (creiamo quindi due sottoinsiemi training e test di 19 attributi). Il modello risultante probabilmente sarà molto complesso ma l’obiettivo è quello di potarlo successivamente. L’albero risultante è il seguente.

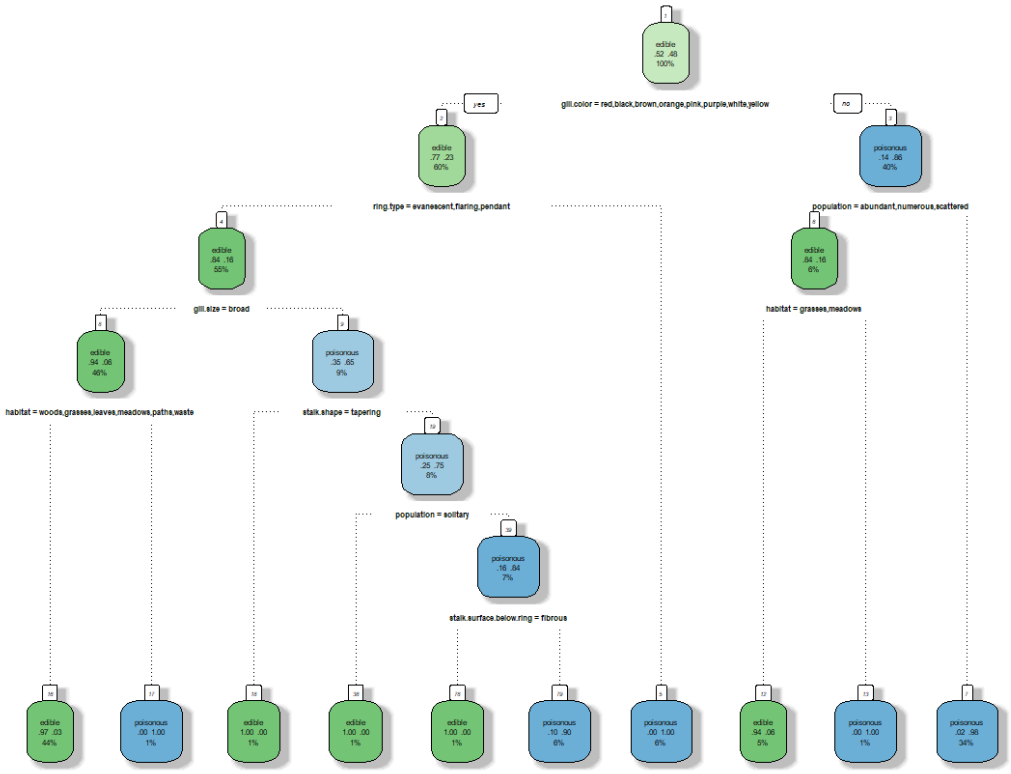


Immagine 9: Albero decisionale allenato considerando 19 attributi (9 splits).

La matrice di confusione computata sulle previsioni di questo albero riporta queste informazioni (e non solo):

```
           Reference
Prediction  edible poisonous
  edible      1231        32
  poisonous     40      1134

               Accuracy : 0.9705
```

Tuttavia siamo interessati ai parametri di complessità di tale albero in modo da decidere a che livello effettuare la potatura, a tal proposito stampiamoli tramite le funzioni *printcp()* e *plotcp().*

```
         CP nsplit rel error   xerror      xstd
1 0.589351      0  1.000000 1.000000 0.0137414
2 0.115974      1  0.410649 0.410649 0.0109592
3 0.087163      2  0.294675 0.294675 0.0096019
4 0.055069      3  0.207513 0.207513 0.0082526
5 0.025894      4  0.152443 0.152443 0.0071770
6 0.015682      6  0.100656 0.100656 0.0059099
7 0.013129      7  0.084974 0.089351 0.0055841
8 0.010576      8  0.071845 0.078775 0.0052571
9 0.010000      9  0.061269 0.070751 0.0049922
```

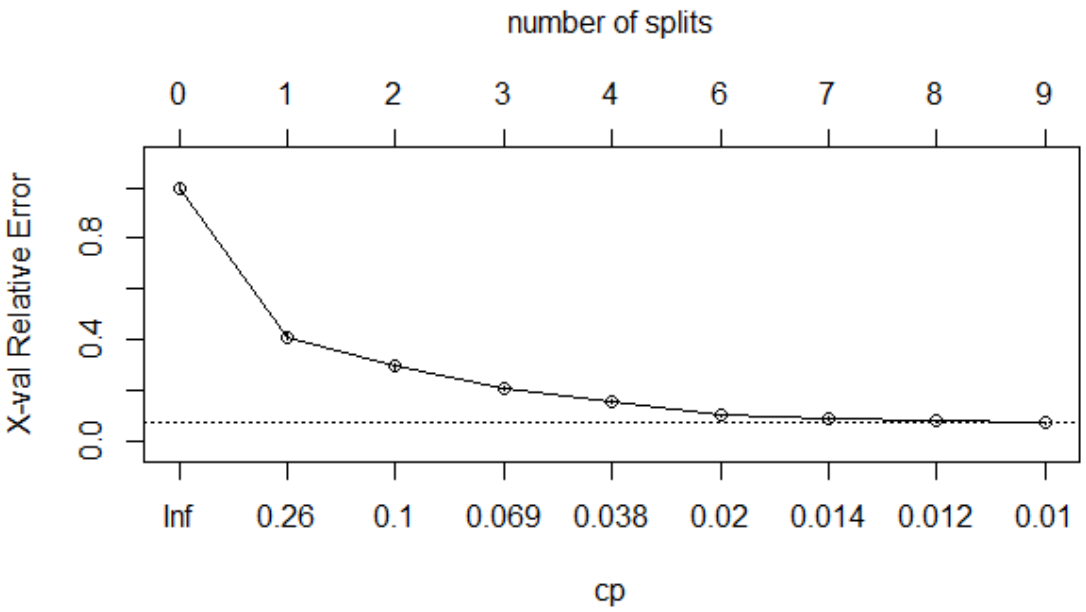


Immagine 10: Plot dei parametri di complessità.

Dalle informazioni riportate qui sopra ricaviamo due soglie di valori per il parametro di complessità.

- La soglia <0.055069, 0.025894> → albero con 4 splits
- La soglia <0.087163, 0.055069> → albero con 3 splits

Abbiamo effettuato inizialmente una potatura utilizzando **0.038** come parametro di complessità, ottenendo un albero con 4 splits (attributi considerati dal modello: *gill.color*, *ring.type*, *population* e *gill.size*). I risultati di questo modello sono riassunti dalla matrice di confusione:

```
            Reference
Prediction  edible poisonous
  edible      1182        81
  poisonous     77      1097
```

accuratezza = 0.9352, *sensitività* = 0.9388 e *specificità* = 0.9312.

Questo modello generalizza meglio rispetto al modello iniziale, tuttavia otteniamo un numero abbastanza elevato di falsi-positivi (81). Come già menzionato all'inizio della relazione il nostro obiettivo è quello di minimizzare i falsi-positivi prediligendo i falsi-negativi (in caso di classificazione errata) in modo da evitare l'avvelenamento.

Procediamo quindi alla potatura dell'albero complesso con parametro di complessità = **0.069** e otteniamo un albero decisionale con 3 splits (attributi considerati dal modello: *gill.color*, *ring.type* e *population).*

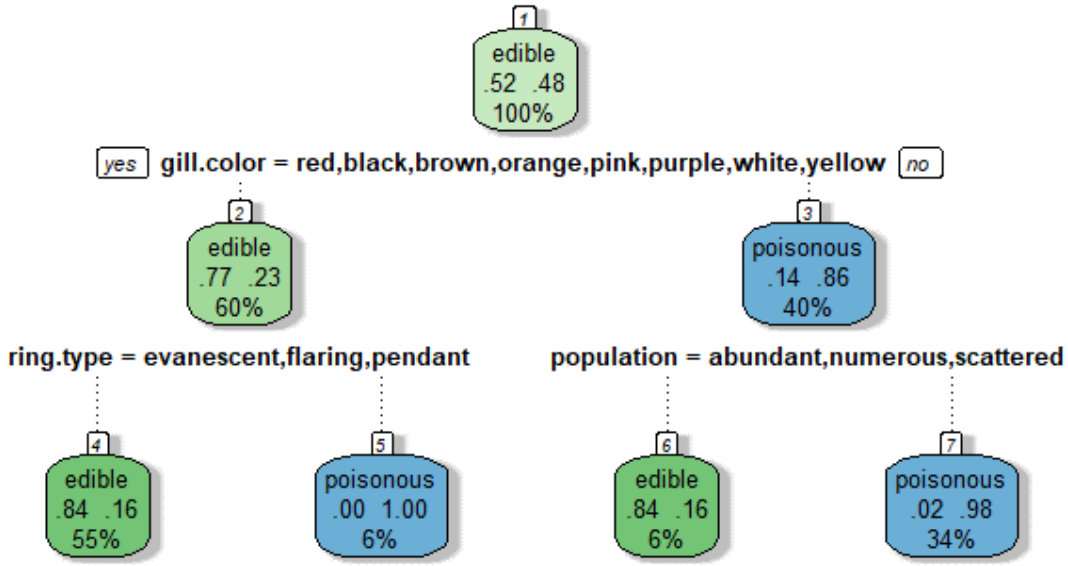


Immagine 11: Albero decisionale potato cp = 0.069.

Risultati sulla predizione di questo modello:

```
           Reference
Prediction  edible poisonous
  edible      1256         7
  poisonous    196       978
```

accuratezza = 0.9167 *, sensitività* = 0.8650 *e specificità* = 0.9929*.*

Come possiamo notare questo modello è ottimale per il nostro caso applicativo in quanto generalizza abbastanza bene, è piuttosto semplice (albero di profondità 2, 3 splits) e presenta una specificità prossima a 1 ovvero classifica poche istanze come falsi-positivi. D'ora in poi considereremo questo modello come quello di riferimento per gli alberi decisionali.

Secondo modello: Macchina a vettori di supporto

Il secondo modello utilizzato è quello della macchina a vettori di supporto (**svm**).

Inizialmente abbiamo eseguito l'allenamento su tutti gli attributi iniziali del training set mantenendo il costo di default (1) e il kernel lineare come metodo di separazione. Questo modello utilizza circa **206** vettori di supporto per effettuare la classificazione delle istanze. Il risultato dell'esecuzione di tale modello sul test set è una separazione perfetta. Segue la matrice di confusione:

```
           Reference
Prediction  edible poisonous
  edible      1263         0
  poisonous      0      1174
```

accuratezza = 1.000 *, sensitività* = 1.000 *e specificità* = 1.000*.*

Ci troviamo ad avere un modello affetto da overfitting, vogliamo verificare come si comporta il modello di SVM considerando solamente gli attributi *odor* e *spore.print.color* come abbiamo operato per gli alberi di decisione.

Il training eseguito considerando questi due attributi produce un modello che considera circa **96** vettori di supporto, quindi un modello più semplice rispetto alla prima SVM. La matrice di confusione (classe positiva = "commestibile") per questo modello è la seguente:

```
           Reference
Prediction  edible poisonous
  edible      1263        11
  poisonous      0      1163
```

accuratezza = 0.9955 *, sensitività* = 1.000 *e specificità* = 0.9906*.*

Anche in questo caso notiamo il modello di Macchina a supporto di vettori sembra essere dominato dagli attributi *odor* e *spore.print.color* quindi operiamo come per gli alberi decisionali, consideriamo i due subset generati in precedenza.

Per determinare il costo ottimo abbiamo eseguito il *tuning* di una SVM considerando il subset del training set con kernel lineare. Ne risulta che il costo migliore per questo caso applicativo è **5**.

Abbiamo quindi effettuato il training della SVM sul subset mantenendo il kernel lineare e impostando il costo a 5. Il modello risultante utilizza **429** vettori di supporto per la classificazione delle istanze. I risultati della predizione sul subset del test set iniziale sono i seguenti:

```
           Reference
Prediction  edible poisonous
  edible      1241        22
  poisonous     17      1157
```

accuratezza = 0.9840 *, sensitività* = 0.9865 *e specificità* = 0.9813*.*

Questi risultati descrivono un modello vicino a quello ideale ma con una complessità molto maggiore rispetto al modello che considerava gli attributi odor e spore.print.color. Abbiamo allenato l' SVM con altre combinazioni di attributi e di kernel ottenendo tuttavia modelli molto complessi, i quali non generalizzano abbastanza sui dati del training set da giustificare un tale aumento nella complessità dell'algoritmo. Considereremo quindi questi ultimi 2 modelli come quelli di riferimento per le SVM.

MISURAZIONE DELLE PERFORMANCE

Precision, Recall e F-Measure dei modelli di riferimento

Per quanto riguarda l'albero di decisione i risultati che ne fuoriescono dalla potatura a 3 split sono validi quando la classe è "commestibile" e "velenoso".

Classe "commestibile":

```
Confusion Matrix and Statistics

          Reference
Prediction  edible poisonous
  edible      1256         7
  poisonous    196       978

               Accuracy : 0.9167
                 95% CI : (0.905, 0.9274)
    No Information Rate : 0.5958
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8322

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.9945
                 Recall : 0.8650
                     F1 : 0.9252
             Prevalence : 0.5958
         Detection Rate : 0.5154
   Detection Prevalence : 0.5183
      Balanced Accuracy : 0.9290

       'Positive' Class : edible
```

Da tali risultati otteniamo le seguenti informazioni:

- *L'accuratezza* del modello è del 0.9167, quindi subisce di una generalizzazione migliore rispetto ai risultati preliminari, in quanto il numero dei FP (falsi-positivi) diminuisce rispettivamente verso 7 funghi velenosi classificati come commestibili (occupando una percentuale bassa) e di 196 funghi FN (falsi negativi) ossia di funghi commestibili apparsi come velenosi.
- Tale cosa può essere di fiducia dal fatto che la precisione della predizione durante il calcolo è quella di 0.9945 (il numero dei TP (veri positivi) ha maggiori dimensioni.

- La *recall* risulta essere abbastanza considerevole (0.8650) in quanto supera la soglia minima del 0.5, quindi vale a dire che ha previsto le osservazioni positive in corrispondenza con la classe “commestibile”.
- La *f-measure* risulta del 0.9252 quindi significa che si ha una distribuzione uniforme.

Classe “velenoso”:

```
Confusion Matrix and Statistics

           Reference
Prediction  edible poisonous
  edible      1256         7
  poisonous    196       978

               Accuracy : 0.9167
                 95% CI : (0.905, 0.9274)
    No Information Rate : 0.5958
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8322

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.8330
                 Recall : 0.9929
                     F1 : 0.9060
             Prevalence : 0.4042
         Detection Rate : 0.4013
   Detection Prevalence : 0.4817
      Balanced Accuracy : 0.9290

       'Positive' Class : poisonous
```

Da tali risultati otteniamo le seguenti informazioni:

- *L’accuratezza* del modello è del 0.9167 ed otteniamo i medesimi risultati della distribuzione dei TP e FP, ma si notano delle differenze
- La *precisione* è scesa considerevolmente a 0.8330. In questo caso quando la classe positiva scelta è “velenoso” comprendiamo che il rapporto della predizione dei positivi si discosta da quello ideale, classificando 196 funghi commestibili come velenosi, tuttavia preferiamo questo tipo di errore (rispetto ai falsi-negativi).
- La *recall* di 0.9929 rimane sempre alta.
- *F-measure* del 0.9060 rimane anche in questo caso alta dal fatto che i dati presentano di una distribuzione uniforme.

Per quanto riguarda il secondo modello (SVM allenata sul subset), otteniamo i seguenti valori:

Classe “commestibile”:

```
Confusion Matrix and Statistics

           Reference
Prediction  edible poisonous
  edible      1241        22
  poisonous     17      1157

               Accuracy : 0.984
                 95% CI : (0.9782, 0.9886)
    No Information Rate : 0.5162
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.968

 Mcnemar's Test P-Value : 0.5218

              Precision : 0.9826
                 Recall : 0.9865
                     F1 : 0.9845
             Prevalence : 0.5162
         Detection Rate : 0.5092
   Detection Prevalence : 0.5183
      Balanced Accuracy : 0.9839

       'Positive' Class : edible
```

Classe “velenoso”:

```
Confusion Matrix and Statistics

           Reference
Prediction  edible poisonous
  edible      1241        22
  poisonous     17      1157

               Accuracy : 0.984
                 95% CI : (0.9782, 0.9886)
    No Information Rate : 0.5162
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.968

 Mcnemar's Test P-Value : 0.5218

              Precision : 0.9855
                 Recall : 0.9813
                     F1 : 0.9834
             Prevalence : 0.4838
         Detection Rate : 0.4748
   Detection Prevalence : 0.4817
      Balanced Accuracy : 0.9839

       'Positive' Class : poisonous
```

Dato che il modello si avvicina a quello ideale, anche i valori di *accuratezza*, *precisione*, *recall* e *F-measure* si allineano a quelli di riferimento (tendono a 1). Evitiamo di analizzare il modello di SVM allenata con i due attributi *odor* e *spore.print.color* in quanto è ancora più allineata al modello ideale.

Curva ROC

Abbiamo voluto valutare l'andamento della curva ROC riferita al modello di SVM allenato utilizzando il subset del training set. Quindi abbiamo allenato il modello calcolando le probabilità delle etichette ed eseguito tale modello sul subset del test set. Abbiamo in seguito estratto le probabilità della classe "velenoso" e disegnato la ROC con la corrispondente *AUC* e otteniamo un'altra conferma sul fatto che questo modello sia molto vicino a quello ideale, infatti l'AUC è prossima a 1.

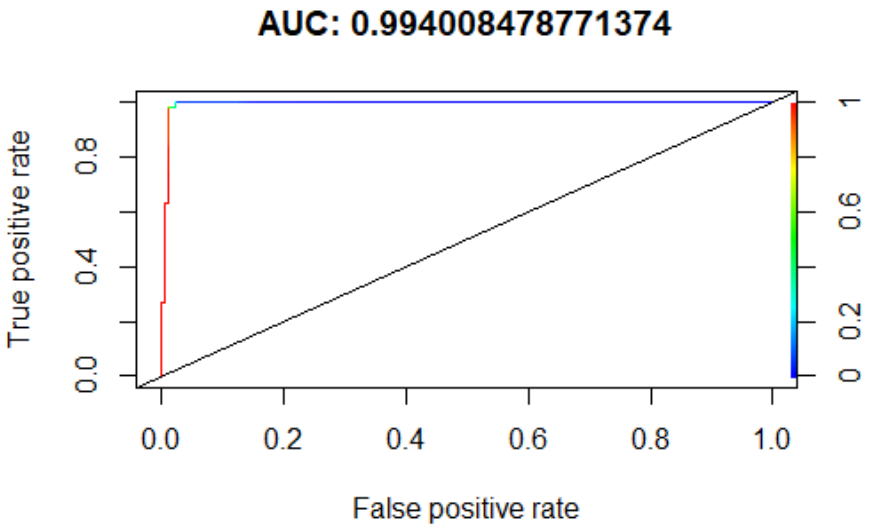


Immagine 12: ROC e corrispondente AUC del modello SVM.

Siamo andati a computare il *cutoff* ottimale per il nostro modello ottenendo un valore di circa **0.77** in relazione ad un'accuratezza di circa 0.98. Segue il plot del cutoff.

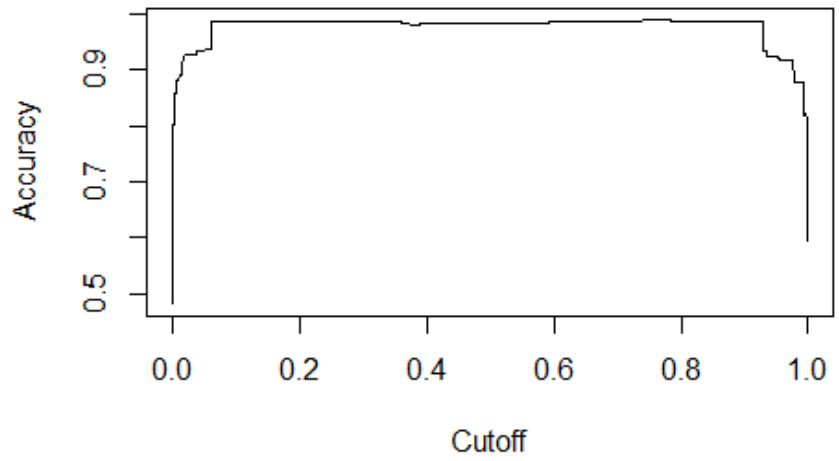


Immagine 12: ROC e corrispondente AUC del modello SVM.

Confronto tra i modelli

In quest'ultima fase abbiamo messo in relazione i modelli per confrontarne le performance. Abbiamo eseguito una 10-fold cross validation con 5 ripetizioni sui seguenti modelli:

- Albero di decisione di riferimento (identificato come rpart)
- SVM allenata sul subset e costo 5 (identificata come svm.model)
- SVM allenata sugli attributi *odor* e *spore.print.color* (identificata come sva.model1)

In seguito all'allenamento dei modelli e alla predizione delle probabilità sulle etichette "commestibile" e "velenoso", siamo andati a computare e visualizzare con un plot unico le ROC relative a ogni modello.

L’immagine 13 rappresenta le tre ROC. In particolare la curva rossa si riferisce al modello *rpart*, la curva arancione si riferisce al modello *svm.model* mentre la curva verde si riferisce al modello *svm.model1*.

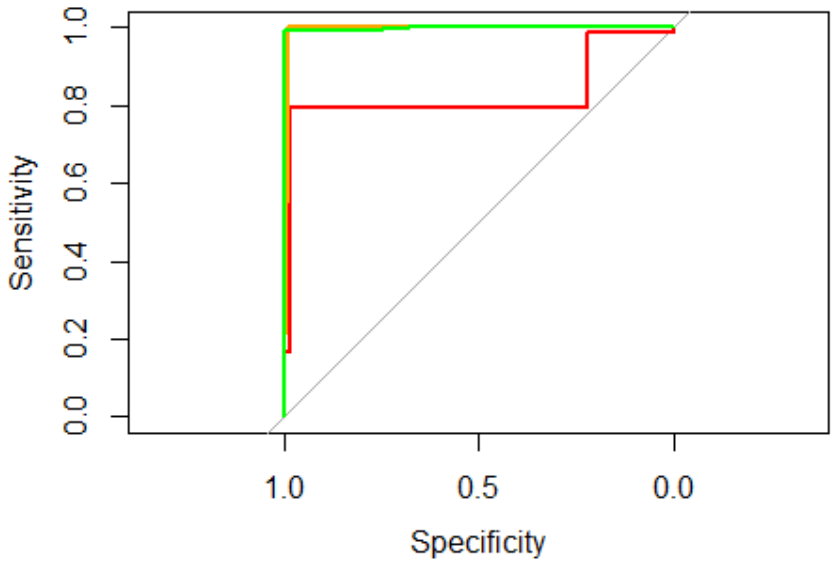


Immagine 13: ROC dei tre modelli.

I corrispondenti valori di AUC:

```
rpart          Area under the curve: 0.9042
svm-model      Area under the curve: 0.9935
svm.model1     Area under the curve: 0.9976
```

Come possiamo notare dalle ROC, i due modelli di SVM si allineano quasi perfettamente pur presentando complessità (vettori di supporto utilizzati) diverse. Mentre il modello basato su albero decisionale sembra generalizzare meglio sulla classificazione, ciò corrisponde alle nostre ipotesi iniziali.

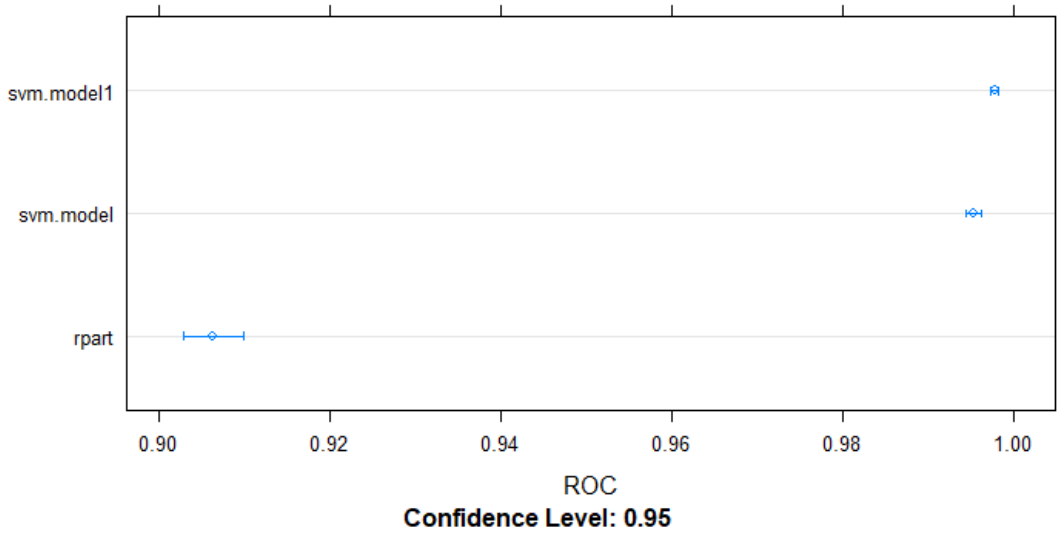


Immagine 14: Intervalli di generalizzazione dei modelli.

L'immagine 14 mostra gli intervalli di generalizzazione (accuratezza) dei modelli con un livello di confidenza pari al 95%.

```
rpart          Accuracy : 0.9167
                 95% CI : (0.905, 0.9274)

svm.model      Accuracy : 0.984
                 95% CI : (0.9782, 0.9886)

Svm.model1     Accuracy : 0.9955
                 95% CI : (0.9919, 0.9977)
```

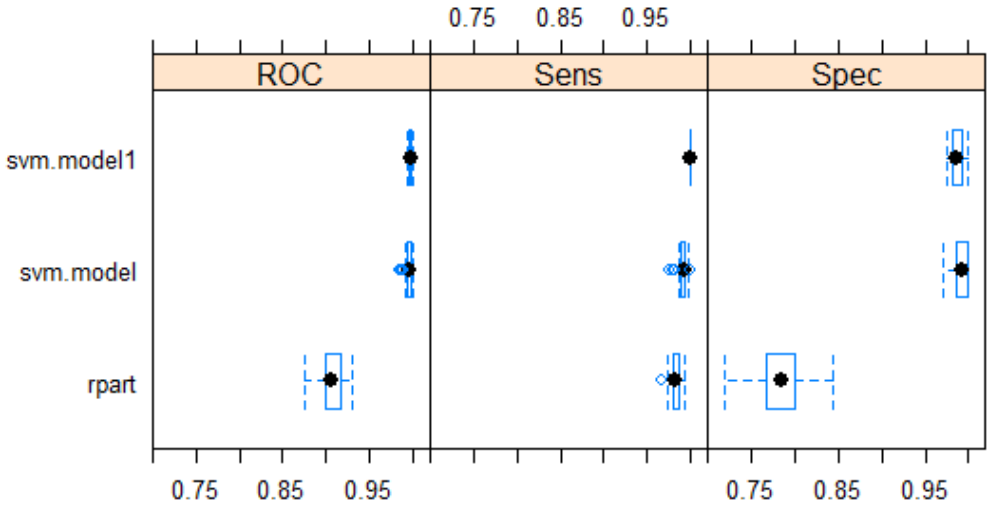


Immagine 15: Plot ROC, Sensitività, Specificità dei modelli.

L’ultimo confronto tra i modelli che abbiamo eseguito è quello sul tempo relativo alla computazione.

```
           Everything FinalModel Prediction
rpart            3.80       0.03         NA
svm.model       81.97       1.68         NA
svm.model1      13.40       0.15         NA
```

Possiamo notare come il modello basato sull’albero decisionale sia molto più performante rispetto a quello basato su SVM, questa osservazione è giustifica anche dal fatto che tale algoritmo generalizza meglio sulle classificazioni. Un’altra osservazione, forse più significativa, è il confronto tra i due modelli di SVM. Seppur la loro performance in relazione alla qualità delle classificazioni sulle istanze sia pressoché identica, il modello allenato con gli attributi *odor* e *spore.print.color* è molto più efficiente in termini di tempo computazionale. Questo fatto spiega che un maggior numero di vettori di supporto considerati dal modello implica una maggiore complessità del modello e computazionale.

CONCLUSIONI

Lo sviluppo di questo progetto ci ha permesso di mettere in pratica e comprendere meglio alcuni concetti studiati solamente nel lato teorico. In particolare ha migliorato il nostro approccio agli algoritmi di apprendimento definendo uno schema procedurale per quanto riguarda l’intero processo di definizione di un modello. Inoltre ha migliorato la nostra familiarità con il linguaggio di programmazione R e il suo editor RStudio.

La rimozione iniziale dei valori mancanti ha aumentato la qualità dei modelli di apprendimento.

I modelli utilizzati hanno presentato un comportamento simile nella classificazione delle istanze, tuttavia il modello basato su alberi decisionali è stato generalizzato con più facilità rispetto alle SVM che tendono a una classificazione ideale.

I risultati ottenuti hanno confermato le nostre ipotesi iniziali tratte dall’analisi del dataset utilizzato. In base ai confronti tra i diversi modelli, considerando qualità delle classificazioni, generalizzazione dal training set e prestazioni nell’esecuzione, riteniamo che il modello basato su albero decisionale sia più adatto alla classificazione delle istanze del nostro dataset.