UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

DIPARTIMENTO DI ECCELLENZA

ANNO ACCADEMICO 2023/2024

Documentazione ML - Mycologist

Nicolò Gallotta (matr. 0512114639)

Repository di GitHub: https://github.com/gnicolo00/Mycologist

Indice

1 Introduzione

Lo studio e la corretta identificazione dei funghi è di fondamentale importanza per preservare la salute umana, mitigando il rischio di possibili **conseguenze nocive associate al consumo di funghi dannosi o addirittura letali**. Inoltre, è altrettanto importante riconoscere l'importanza del mantenimento della vita fungina nell'ecosistema, sensibilizzando pratiche sostenibili nella raccolta.
Evitare eventuali avvelenamenti fungini rappresenta una sfida persistente per i fungaioli appassionati e per i micologi, poiché il riconoscimento delle varie specie e delle caratteristiche fungine richiede una conoscenza approfondita. Questa problematica sottolinea la necessità di sviluppare strumenti avanzati, affidabili e intelligenti per la classificazione delle specie fungine velenose. In questo contesto, questo progetto si propone di colmare questa lacuna.

2 Definizione del problema

2.1 Obiettivi

L'obiettivo di questo progetto è sviluppare un modello di Machine Learning che possa analizzare dei dati riguardanti le caratteristiche di funghi, in particolare della famiglia Agaricus e Lepiota, per **identificare e distinguere i funghi velenosi dai funghi commestibili**. Questa iniziativa riveste particolare importanza, poiché la corretta identificazione dei funghi è essenziale per evitare gravi rischi per la salute umana.
Tra i dati a nostra disposizione abbiamo caratteristiche specifiche come la forma del cappello, il colore del cappello, la forma del gambo, la presenza di contusioni e altre più generiche come l'habitat, la disposizione nel bosco, e molte altre.

2.2 Analisi del problema

Il modello è progettato per condurre un'analisi volta a identificare funghi velenosi sulla base di specifiche caratteristiche. Pertanto, si tratta di un'istanza di un problema di apprendimento supervisionato, più precisamente di **classificazione binaria**. Saranno impiegate e confrontate diverse tecniche di Machine Learning al fine di individuare la più efficace, basandoci su diverse metriche di valutazione.

3 Analisi dei dati

3.1 Acquisizione del dataset

I dati utilizzati per l'addestramento del modello provengono dal sito web UC Irvine Machine Learning Repository. In particolare, il dataset utilizzato è reperibile al seguente link: `https://archive.ics.uci.edu/dataset/73/mushroom`.

3.2 Struttura del modello dei dati

Il dataset è composto da **8124 righe** e **23 colonne** e, per ognuna delle feature, sono presenti delle **variabili categoriche**. In particolare, i valori che una caratteristica specifica (e.g. forma del cappello) può assumere sono riportati con una singola lettera, ognuna con il proprio significato. Scendendo più nel dettaglio, le seguenti sono le caratteristiche a disposizione, con i relativi valori:

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

4 Analisi ed Elaborazione dei dati

4.1 Variabile dipendente

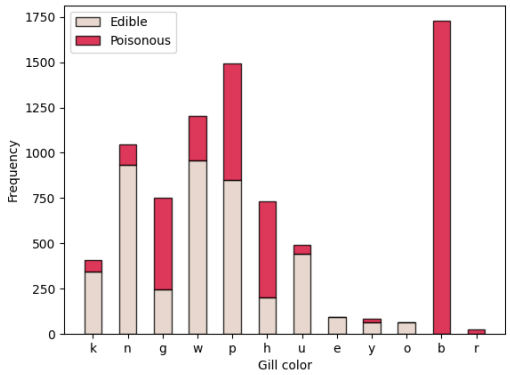
Come menzionato precedentemente, nel dataset sono presenti 23 colonne. In particolare, tra queste vi troviamo la **variabile dipendente** il cui valore dovrà essere predetto dal modello, ovvero la variabile che determina la tossicità o la commestibilità del fungo. Questa caratteristica assume due valori: **'p'** ed **'e'**, rispettivamente '*poisonous*' ed '*edible*'.

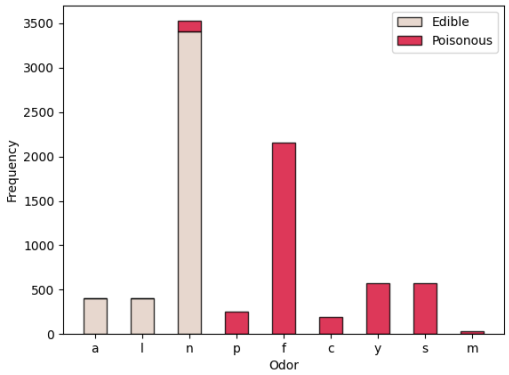
4.2 Valori mancanti

Sono stati precedentemente elencati i valori che ogni feature può assumere all'interno del dataset. Tra questi, possiamo notare come la colonna '*stalk-root*' può assumere un particolare valore, ovvero **'?'**, il che indica la mancanza di valore. In questo dataset, infatti, i valori mancanti sono stati rappresentati con un punto interrogativo anziché essere lasciati nulli, e sono **2480 in totale**. Fortunatamente, '*stalk-root*' è l'unica colonna che presenta valori mancanti, il che ci facilita tenere sotto controllo tale problematica.

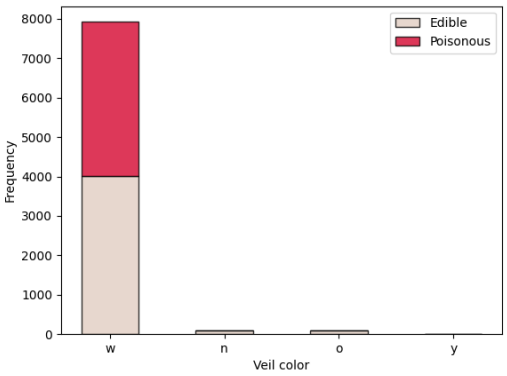
4.3 Distribuzioni

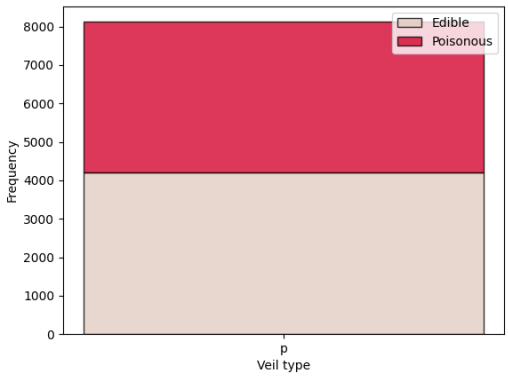
Di seguito sono riportate le distribuzioni più interessanti rispetto alla variabile dipendente:





Notiamo come, in questi due istogrammi, ci sono dei valori determinanti e forse **troppo correlati alla variabile dipendente**, soprattutto per quanto riguarda '*odor*'. Ciò suggerisce che molto probabilmente queste variabili potrebbero creare dei bias nel modello o, addirittura, rappresentare un fenomeno di Data Leakage.





Notiamo come, in questi altri due istogrammi, ci sono dei valori incredibilmente predominanti e addirittura singoli. Sotto un certo punto di vista, possiamo considerare queste features come **costanti**, poiché il loro valore cambia in maniera estremamente rara o non cambia affatto. Ciò ci suggerisce che tali features non saranno di alcuna utilità al modello, poiché forniscono una potenza predittiva troppo bassa.

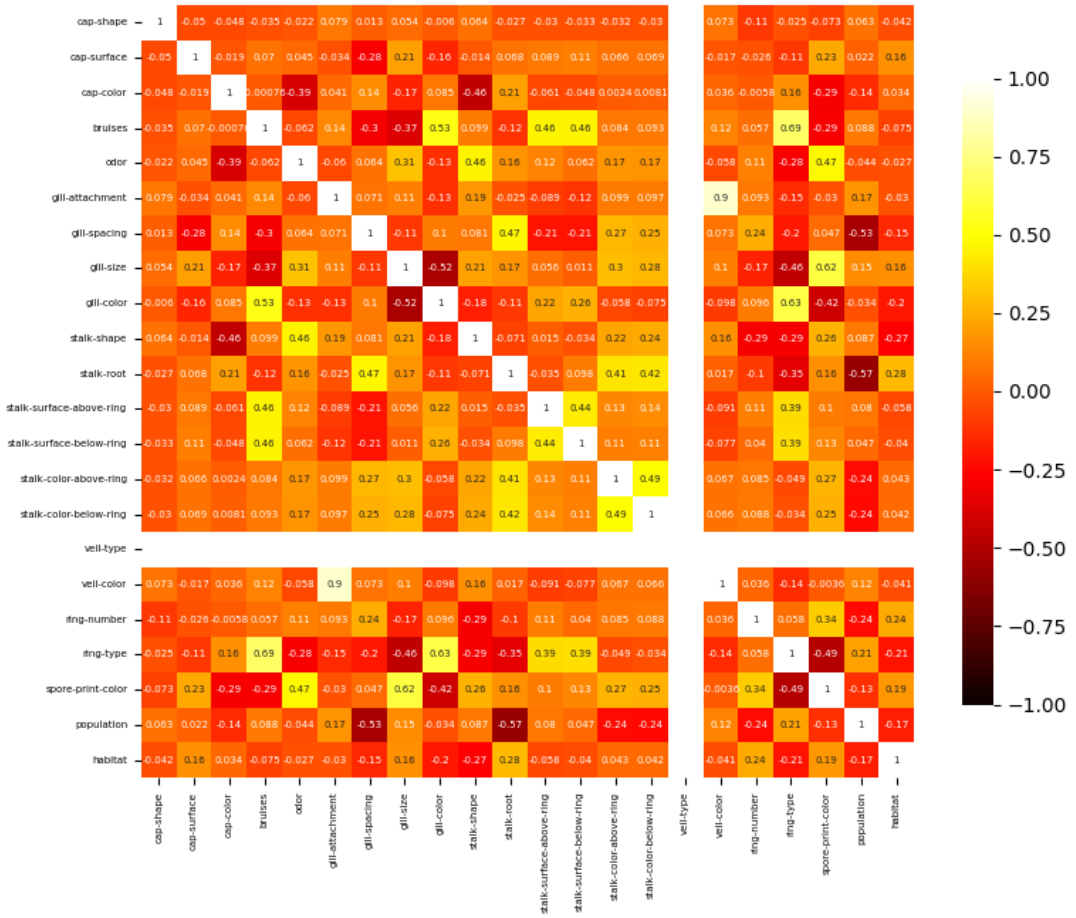
4.4 Encoding delle variabili categoriche

Come visto in precedenza, le features del dataset assumono valori categorici e sotto forma di caratteri singoli. Tuttavia, poiché i modelli di Machine Learning e le loro funzioni lavorano meglio (o solo) con i numeri, è necessario codificare questi caratteri. In particolare, è stato utilizzato il **Label Encoding**: una tecnica di trasformazione di variabili categoriche in numeri interi positivi. In questo processo, le categorie uniche in una variabile categorica vengono assegnate a interi in modo sequenziale. Ad esempio, la nostra variabile categorica '*veil-color*' assume i valori 'n', 'o', 'w' e 'y' che il Label Encoder potrebbe assegnare loro i valori numerici 0, 1, 2 e 3 rispettivamente. Il seguente è un estratto del risultato del Label Encoding applicato al dataset:

```
class
p → 1, e → 0

cap-shape
x → 5, b → 0, s → 4, f → 2, k → 3, c → 1

cap-surface
s → 2, y → 3, f → 0, g → 1

cap-color
n → 4, y → 9, w → 8, g → 3, e → 2, p → 5, b → 0, u → 7, c → 1, r → 6

bruises
t → 1, f → 0

odor
p → 6, a → 0, l → 3, n → 5, f → 2, c → 1, y → 8, s → 7, m → 4

gill-attachment
f → 1, a → 0

gill-spacing
c → 0, w → 1

gill-size
n → 1, b → 0

gill-color
k → 4, n → 5, g → 2, p → 7, w → 10, h → 3, u → 9, e → 1, b → 0, r → 8, y → 11, o → 6

stalk-shape
e → 0, t → 1
```

4.5 Correlazioni tra le features

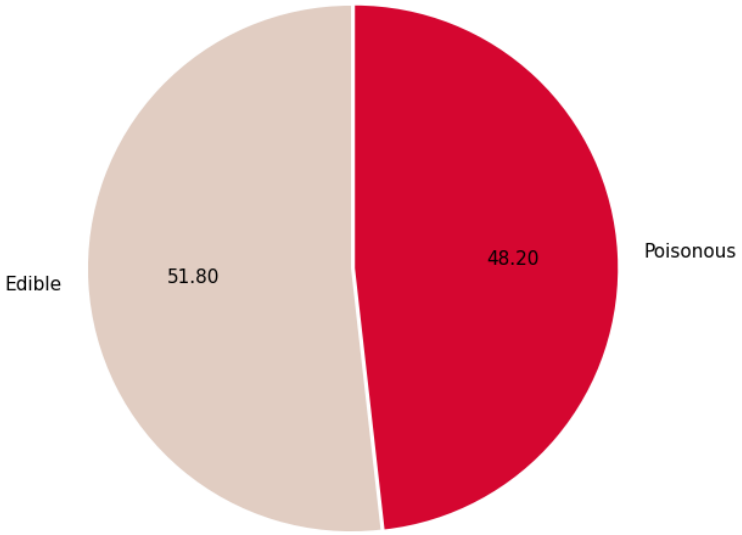
Di seguito sono riportate le correlazioni tra le features attraverso una ***heatmap***:



Notiamo come la riga e la colonna relativa alla feature '*veil-type*' è vuota. Questo è dovuto al fatto che essa assume un singolo valore, il che comporta una correlazione inesistente con le altre. Un valore del genere, nel contesto delle heatmap, è definito '*bad-value*'. Questo sottolinea la necessità di eliminare tale feature dal nostro dataset.

4.6 Bilanciamento

A questo punto, possiamo verificare il bilanciamento della classe da predire, osservando il seguente grafico a torta:



Come si può osservare, il dataset si presenta **quasi perfettamente bilanciato**, con una leggera prevalenza nelle classi in cui i funghi sono classificati come commestibili. Questo equilibrio fornirà una base solida per l'addestramento del nostro modello, consentendogli di apprendere in modo accurato dalle diverse categorie presenti nel dataset. Inoltre, questo ci facilita il lavoro poiché non sarà necessario toccare né la classe minoritaria né la classe maggioritaria.

4.7 Eliminazione delle features inadatte

Per le motivazioni precedentemente discusse, possiamo eliminare dal dataset le seguenti features: '*veil-color*', '*gill-attachment*', '*veil-type*' e '*odor*'. Le prime due presentano una distribuzione completamente sbilanciata, dove viene assunto un solo valore più del 99% delle volte. La terza assume un singolo valore, non fornendo alcun vantaggio per il modello e creando rumore. L'ultima, infine, presenta una correlazione troppo elevata con la variabile dipendente.

4.8 Imputazione dei dati mancanti

Poiché il numero di righe in cui i sono presenti valori mancanti è elevato rispetto al numero totale di campioni nel dataset (2480 su 8124) e poiché la radice potrebbe rappresentare una caratteristica importante per l'identificazione di funghi velenosi, è stata presa la decisione di evitare di eliminare le righe o la colonna '*stalk-root*' in cui sono presenti i valori mancanti. Piuttosto, è stata effettuata un'imputazione attraverso l'utilizzo della **moda** all'interno della colonna stessa.

5 Addestramento e Valutazione

Una volta che i dati sono stati individuati e resi pronti all'utilizzo, non ci rimane che impiegarli per l'addestramento del nostro modello. Poiché stiamo affrontando un'istanza di un problema di classificazione binaria, è stato deciso di utilizzare, testare e confrontare due algoritmi particolarmente adatti a lavorare con valori discreti, come nel nostro caso: ***Naive Bayes*** e ***Logistic Regression***. Le metriche utilizzate per la valutazione dei modelli sono le seguenti: ***Accuracy***, ***Precision***, ***Recall*** ed ***F1 Score***. In particolare, è stata data priorità al modello con la **Recall più alta**, poiché questo significherebbe il minor numero di falsi negativi possibile. Questi, infatti, sono di estrema importanza dal momento che rappresentano le situazioni in cui un fungo velenoso è erroneamente classificato come commestibile, il che può portare a conseguenze estremamente gravi.
Ovviamente, prima di procedere, il dataset è stato **suddiviso in set di training e set di testing** attraverso uno split 80-20. Ciò ci ha lasciato con 6499 campioni per il training e 1625 campioni per il testing.

5.1 Naive Bayes

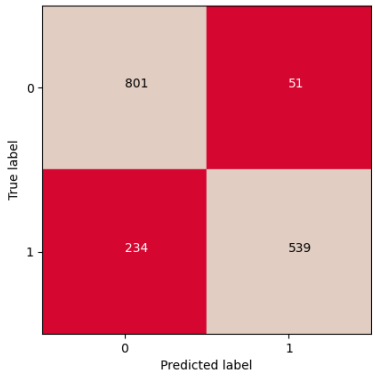
Per quanto riguarda il modello *Naive Bayes*, i valori ottenuti durante il training relativi ad ognuna delle metriche sono i seguenti:

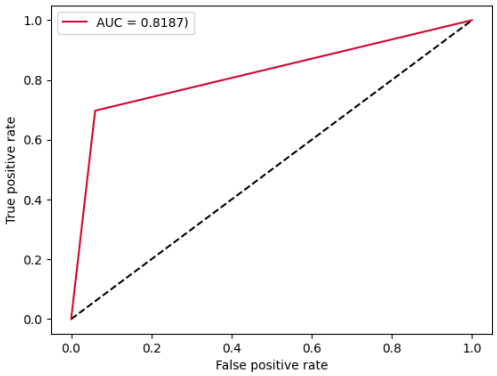
- **Accuracy**: 0.82;
- **Precision**: 0.92;
- **Recall**: 0.69;
- **F1 Score**: 0.79.

I valori ottenuti durante il testing, invece, sono i seguenti:

- **Accuracy**: 0.82;
- **Precision**: 0.91;
- **Recall**: 0.70;
- **F1 Score**: 0.79.

Di seguito sono riportate la *confusion matrix* e la *ROC Curve*:





5.2 Logistic Regression

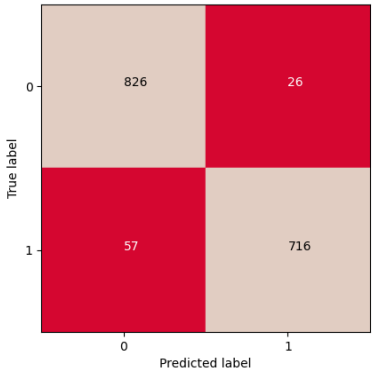
Per quanto riguarda il modello *Logistic Regression*, i valori ottenuti durante il training relativi ad ognuna delle metriche sono i seguenti:

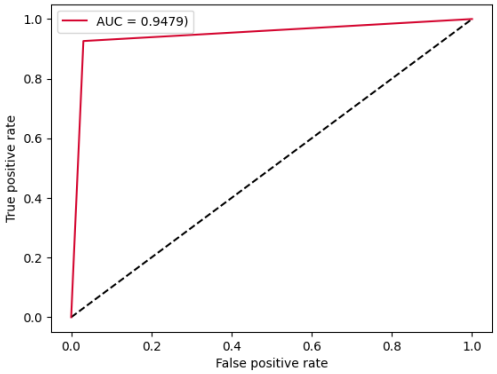
- **Accuracy**: 0.95;
- **Precision**: 0.96;
- **Recall**: 0.93;
- **F1 Score**: 0.94.

I valori ottenuti durante il testing, invece, sono i seguenti:

- **Accuracy**: 0.95;
- **Precision**: 0.96;
- **Recall**: 0.93;
- **F1 Score**: 0.95.

Di seguito sono riportate la *confusion matrix* e la *ROC Curve*:





6 Scelta del modello

Osservando i risultati sopra riportati, notiamo che **il *Logistic Regression* ha dimostrato una performance maggiore rispetto a *Naive Bayes*** nel nostro contesto specifico, sia durante il training che durante il testing. La sue elevate metriche durante entrambe le fasi indicano che il modello è in grado di fare previsioni accurate su entrambe le classi, riducendo al minimo sia i falsi positivi che i falsi negativi.
Per queste motivazioni, il *Logistic Regression* è il modello più adatto alla risoluzione del nostro problema.

7 Conclusioni

Concluso il processo di sviluppo e la selezione del modello, posso affermare senza dubbio di essere soddisfatto dei risultati ottenuti. Nonostante le difficoltà incontrate, ho non solo implementato con successo i concetti appresi durante le lezioni della materia, ma ho anche acquisito nuove competenze che ho successivamente applicato. L'esperienza mi ha fornito una più ampia comprensione della complessa struttura dei modelli, le diverse tecniche di encoding, e altre nozioni fondamentali nell'ambito dell'apprendimento automatico.
Questo progetto non ha solo consolidato le mie competenze esistenti, ma ha anche ampliato il mio bagaglio di conoscenze, avvicinandomi di più al raggiungimento dei miei obiettivi.