Análise Preditiva de Cogumelos Comestíveis e Venenosos

Bruno Castro Tomaz[1], Tomás Fiorelli Barbosa[1]

[1]Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie
Rua da Consolação, 930 – Consolação – 01302 – 907 – São Paulo – SP – Brazil

10389988@mackenzista.com.br, 10395687@mackenzista.com.br

Resumo. *Este artigo descreve os resultados obtidos após a realização de análise preditiva, utilizando algoritmos de aprendizagem de máquina Supervisionados, em um conjunto de dados com informações a respeito de cogumelos – formato, coloração, tamanhos e entre outros. O objetivo da análise é prever quais cogumelos são comestíveis e quais são venenosos (classes preditas).*

1. Introdução

a) Contextualização

A classificação de cogumelos entre comestíveis e venenosos é um problema de grande relevância para a segurança alimentar e a saúde pública. Há mais de 10 mil espécies de cogumelos conhecidas, das quais algumas são altamente tóxicas e podem causar sérios riscos à saúde humana. O reconhecimento manual de espécies comestíveis requer conhecimentos especializados, e erros podem ser fatais. Nesse cenário, o uso de tecnologias de machine learning (ML) se mostra uma alternativa promissora, possibilitando a automatização e a aceleração desse processo de identificação. Utilizando um dataset estruturado contendo características físicas de cogumelos, como cor, formato, tamanho, entre outros, pode-se construir modelos preditivos capazes de identificar automaticamente se um cogumelo é venenoso ou seguro para consumo.

b) Justificativa

O uso de algoritmos de aprendizado supervisionado para classificação de cogumelos se justifica pela capacidade dessas técnicas de lidar com grandes volumes de dados e gerar predições com alta precisão, auxiliando na detecção de padrões não triviais. A aplicação de machine learning em áreas biológicas, como a micologia, oferece não apenas benefícios práticos imediatos, como a redução do risco de intoxicações, mas também a oportunidade de explorar novos modelos preditivos que possam ser aplicados a outras áreas da biologia. Além disso, a automação deste processo diminui a dependência de especialistas, tornando a identificação de cogumelos comestíveis acessível a um público mais amplo.

c) Objetivo

O objetivo deste trabalho é aplicar algoritmos de machine learning supervisionado em um dataset de cogumelos, com o intuito de desenvolver modelos capazes de prever, com alta precisão, se um cogumelo é comestível ou venenoso. Pretende-se comparar diferentes algoritmos supervisionados (como árvores de decisão, k-Nearest Neighbors, e Random Forest) para identificar qual deles apresenta o melhor desempenho no problema de classificação. O foco será em métricas de acurácia, precisão e recall, de modo a garantir a robustez do modelo na identificação correta das classes.

d) Opção Escolhida

A opção escolhida para nosso projeto é “Opção Framework”.

2. Descrição do Problema

O problema abordado neste projeto consiste em prever se um cogumelo é comestível ou venenoso com base em suas características físicas e sensoriais. Para essa tarefa, foi utilizado o dataset "Mushroom Overload” disponível no Kaggle, que contém informações detalhadas de 21 variáveis sobre cogumelos, incluindo aspectos como formato do chapéu, superfície do chapéu, cor do chapéu e entre outras características físicas.

Os principais desafios do problema de classificação incluem:

1. Multidimensionalidade: o dataset possui várias características categóricas, algumas com muitos valores distintos. A escolha do método de codificação adequado para essas variáveis é essencial para a construção de modelos eficazes.

2. Correlação entre características: algumas variáveis podem estar fortemente correlacionadas, o que pode impactar o desempenho dos algoritmos de aprendizado de máquina, criando a necessidade de técnicas de seleção de características.

3. Desequilíbrio de classes: embora o dataset seja balanceado em termos de número de amostras comestíveis e venenosas, garantir a precisão do modelo tanto para cogumelos comestíveis quanto para venenosos é essencial para evitar falsos negativos, que poderiam levar a graves consequências de saúde.

4. Interpretação de resultados: além da precisão, é importante que o modelo seja interpretável, dado que o objetivo é fornecer uma ferramenta confiável para auxiliar na identificação de cogumelos potencialmente perigosos.

3. Descrição do Dataset

O dataset é composto por mais de 6 milhões de amostras, sendo que cada amostra representa uma observação de cogumelo, categorizada como comestível (edible) ou venenoso (poisonous). As variáveis do dataset são majoritariamente categóricas, descritas por rótulos que codificam diferentes propriedades de cada cogumelo. A classe alvo para predição é a variável que indica se o cogumelo é comestível ou venenoso, sendo representada por um rótulo binário.

A seguir, uma explicação detalhada de cada campo/coluna do dataset:

n = categórico | m = numérico

1. **cap-diameter (m)**: número decimal em cm
2. **cap-shape (n)**: sino=b, cônico=c, convexo=x, plano=f, afundado=s, esférico=p, outros=o
3. **cap-surface (n)**: f fibroso=i, sulcos=g, escamoso=y, liso=s, seco=d, brilhante=h, coriáceo=l, sedoso=k, pegajoso=t, enrugado=w, carnoso=e
4. **cap-color (n)**: marrom=n, bege=b, cinza=g, verde=r, rosa=p, roxo=u, vermelho=e, branco=w, amarelo=y, azul=l, laranja=o, preto=k
5. **does-bruise-bleed (n)**: machuca-ou-sangra=t, não=f
6. **gill-attachment (n)**: aderida=a, anexada=x, decorrente=d, livre=e, sinuada=s,

poros=p, nenhuma=f, desconhecido=?

7. **gill-spacing (n)**: próxima=c, distante=d, nenhuma=f
8. **gill-color (n)**: cor-do-chapéu + nenhuma=f
9. **stem-height (m)**: número decimal em cm
10. **stem-width (m)**: número decimal em mm
11. **stem-root (n)**: bulboso=b, inchado=s, em-clava=c, em-copo=u, igual=e, rizomorfos=z, enraizado=r
12. **stem-surface (n)**: superfície-do-chapéu + nenhuma=f
13. **stem-color (n)**: cor-do-chapéu + nenhuma=f
14. **veil-type (n)**: parcial=p, universal=u
15. **veil-color (n)**: cor-do-chapéu + nenhuma=f
16. **has-ring (n)**: anel=t, nenhum=f
17. **ring-type (n)**: em-teia=c, evanescente=e, esvoaçante=r, sulcado=g, grande=l, pendente=p, envolvente=s, em-zona=z, escamoso=y, móvel=m, nenhum=f, desconhecido=?
18. **spore-print-color (n)**: cor do chapeú
19. **habitat (n)**: gramados=g, folhas=l, campos=m, caminhos=p, charnecas=h, urbano=u, resíduos=w, bosques=d
20. **season (n)**: primavera=s, verão=u, outono=a, inverno=w
21. **class (n)**: e=comestível, p=venenoso

Para o desenvolvimento da primeira parte deste trabalho, foram utilizadas as seguintes bibliotecas no ambiente Jupyter Notebook:

```
2.1.1  <numpy>
2.2.2  <pandas>
1.5.2  <scikit-learn>
0.13.2 <seaborn>
0.14.3 <statsmodels>
3.9.2  <matplotlib>
```

Como descrito anteriormente, as variáveis presentes neste *dataset* são principalmente categóricas e, conforme a imagem abaixo, foi utilizado o comando *dtypes* para fazer essa identificação através do código:

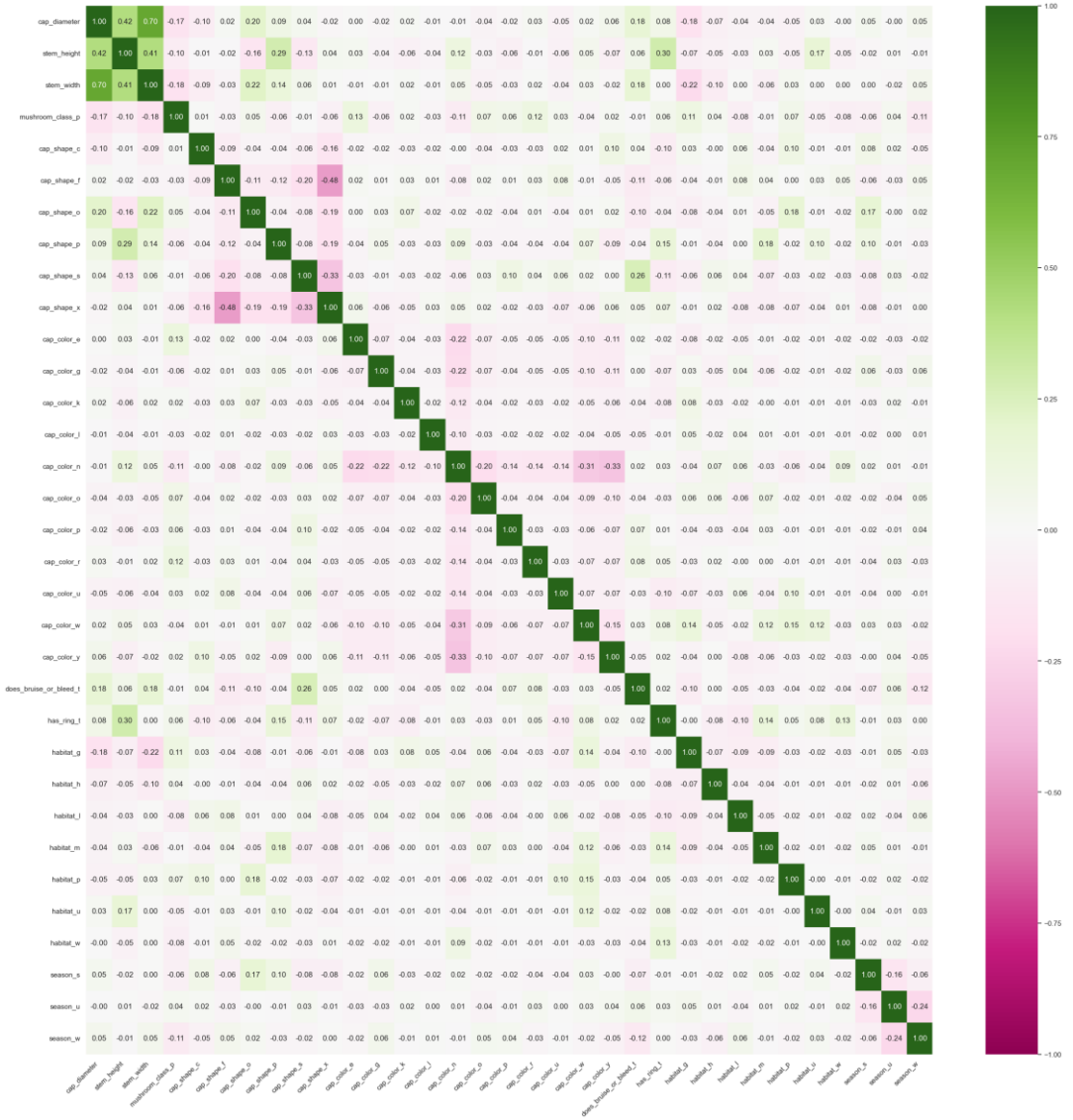
```
mushroom_class           object
cap_diameter            float64
cap_shape                object
cap_surface              object
cap_color                object
does_bruise_or_bleed     object
gill_attachment          object
gill_spacing             object
gill_color               object
stem_height             float64
stem_width              float64
stem_root                object
stem_surface             object
stem_color               object
veil_type                object
veil_color               object
has_ring                 object
ring_type                object
spore_print_color        object
habitat                  object
season                   object
dtype: object
```

Quanto a remoção de atributos desnecessários e linhas problemáticas, incialmente, utilizou-se do comando *isnull* para verificarmos a presença de valores nulos em cada coluna, e pudemos perceber que algumas delas apresentaram um número elevado de dados omitidos (quase 75% para uma única coluna, por exemplo). Além disso, olhando a descrição dos atributos, foi identificado a repetição da coloração do cogumelo em diversas partes (*gill_color* e *stem_color* são atributos baseados em *cap_color* e, portanto, também foram removidas. Outro valor que foi interpretado pelo grupo como inconsistente foi a presença de uma única linha com *stem_width* (largura do caule) possuia tamanho negativo.

	cap_diameter	stem_height	stem_width
count	6.723116e+06	6.723116e+06	6.723116e+06
mean	6.792650e+00	6.697878e+00	1.236342e+01
std	5.279232e+00	3.300607e+00	9.967683e+00
min	2.200000e-01	0.000000e+00	-6.400000e-01
25%	3.520000e+00	4.710000e+00	5.470000e+00
50%	5.960000e+00	6.010000e+00	1.039000e+01
75%	8.600000e+00	7.790000e+00	1.672000e+01
max	6.689000e+01	3.770000e+01	1.186800e+02

```
mushroom_class               0
cap_diameter                 0
cap_shape                    0
cap_surface            1579337
cap_color                    0
does_bruise_or_bleed         0
gill_attachment        1066721
gill_spacing           2800709
gill_color                   0
stem_height                  0
stem_width                   0
stem_root              5763499
stem_surface           4263506
stem_color                   0
veil_type              6367343
veil_color             5893024
has_ring                     0
ring_type               276729
spore_print_color      6049365
habitat                      0
season                       0
dtype: int64
```

Após a remoção desses dados, realizou-se a codificação das variáveis categóricas em numéricas através do método *One-Hot Enconding*, transformando os 10 atributos iniciais (incluindo o alvo) em 33 atributos. Com isso, foi obtido um mapa de calor (OBS: por possuir muitas variáveis, é recomendável abrir o arquivo de imagem salvo ao executar o código - *mushroom_onehotenconding.png*.



A partir deste ponto, obteve-se os resultados gerados a partir da criação dos modelos de predição por regressão linear e logística. Contudo, não foram obtidos valores satisfatórios para o desenvolvimento de um modelo aceitável, pois, para ambos, a métrica R^2 obteve um resultado muito baixo.

Regressão Linear | R^2 e R^2-ajustado = 0.158

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       mushroom_class_p   R-squared:                       0.158
Model:                           OLS   Adj. R-squared:                  0.158
Method:                Least Squares   F-statistic:                 4.071e+04
Date:               Sun, 29 Sep 2024   Prob (F-statistic):               0.00
Time:                       21:14:40   Log-Likelihood:            -4.2710e+06
No. Observations:            6723115   AIC:                         8.542e+06
Df Residuals:                6723083   BIC:                         8.543e+06
Df Model:                         31
Covariance Type:            nonrobust
==============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
Intercept                                     0.5816      0.001    399.066      0.000       0.579       0.584
cap_shape_c[T.True]                          -0.1800      0.001   -148.998      0.000      -0.182      -0.178
cap_shape_f[T.True]                          -0.2223      0.001   -307.424      0.000      -0.224      -0.221
cap_shape_o[T.True]                           0.0103      0.001      8.781      0.000       0.008       0.013
cap_shape_p[T.True]                          -0.2135      0.001   -198.697      0.000      -0.216      -0.211
cap_shape_s[T.True]                          -0.2389      0.001   -278.923      0.000      -0.241      -0.237
cap_shape_x[T.True]                          -0.2458      0.001   -365.493      0.000      -0.247      -0.245
cap_color_e[T.True]                           0.5419      0.001    377.636      0.000       0.539       0.545
cap_color_g[T.True]                           0.1778      0.001    123.658      0.000       0.175       0.181
cap_color_k[T.True]                           0.3045      0.002    171.714      0.000       0.301       0.308
cap_color_l[T.True]                           0.1603      0.002     81.900      0.000       0.156       0.164
cap_color_n[T.True]                           0.2332      0.001    180.220      0.000       0.231       0.236
cap_color_o[T.True]                           0.4409      0.001    301.846      0.000       0.438       0.444
cap_color_p[T.True]                           0.4869      0.002    295.515      0.000       0.484       0.490
cap_color_r[T.True]                           0.6001      0.002    367.837      0.000       0.597       0.603
cap_color_u[T.True]                           0.3886      0.002    238.194      0.000       0.385       0.392
cap_color_w[T.True]                           0.2201      0.001    160.803      0.000       0.217       0.223
cap_color_y[T.True]                           0.3061      0.001    226.843      0.000       0.303       0.309
does_bruise_or_bleed_t[T.True]                0.0153      0.001     29.953      0.000       0.014       0.016
has_ring_t[T.True]                            0.1033      0.000    224.742      0.000       0.102       0.104
habitat_g[T.True]                             0.1020      0.001    179.200      0.000       0.101       0.103
habitat_h[T.True]                             0.0736      0.001     72.859      0.000       0.072       0.076
habitat_l[T.True]                            -0.0685      0.001    -83.262      0.000      -0.070      -0.067
habitat_m[T.True]                            -0.0939      0.001   -108.212      0.000      -0.096      -0.092
habitat_p[T.True]                             0.3457      0.002    143.268      0.000       0.341       0.350
habitat_u[T.True]                            -0.3771      0.004    -92.374      0.000      -0.385      -0.369
habitat_w[T.True]                            -0.5656      0.002   -238.712      0.000      -0.570      -0.561
season_s[T.True]                             -0.2096      0.001   -226.923      0.000      -0.211      -0.208
season_u[T.True]                             -0.0165      0.000    -43.036      0.000      -0.017      -0.016
season_w[T.True]                             -0.1759      0.001   -264.489      0.000      -0.177      -0.175
stem_height                                  -0.0069   6.99e-05    -98.270      0.000      -0.007      -0.007
stem_width                                   -0.0063   2.27e-05   -279.159      0.000      -0.006      -0.006
==============================================================================
Omnibus:                 47164025.602   Durbin-Watson:                   0.115
Prob(Omnibus):                 0.000   Jarque-Bera (JB):           604923.241
Skew:                         -0.115   Prob(JB):                         0.00
Kurtosis:                      1.549   Cond. No.                         432.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Regressão Logística | Pseudo $R^2 = 0.1285$

```
                           Logit Regression Results
==============================================================================
Dep. Variable:       mushroom_class_p   No. Observations:              6723115
Model:                         Logit   Df Residuals:                  6723083
Method:                          MLE   Df Model:                           31
Date:                Sun, 29 Sep 2024   Pseudo R-squ.:                 0.1285
Time:                        20:41:46   Log-Likelihood:          -4.0351e+06
converged:                      False   LL-Null:                 -4.6299e+06
Covariance Type:            nonrobust   LLR p-value:                    0.000
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
Intercept                      0.4383      0.008     56.381      0.000       0.423       0.454
cap_shape_c[T.True]           -0.9334      0.006   -157.998      0.000      -0.945      -0.922
cap_shape_f[T.True]           -1.1126      0.004   -293.776      0.000      -1.120      -1.105
cap_shape_o[T.True]            0.0219      0.006      3.620      0.000       0.010       0.034
cap_shape_p[T.True]           -1.0659      0.005   -201.849      0.000      -1.076      -1.056
cap_shape_s[T.True]           -1.1999      0.004   -274.979      0.000      -1.208      -1.191
cap_shape_x[T.True]           -1.2237      0.004   -342.787      0.000      -1.231      -1.217
cap_color_e[T.True]            2.5821      0.008    333.552      0.000       2.567       2.597
cap_color_g[T.True]            0.8613      0.008    114.034      0.000       0.846       0.876
cap_color_k[T.True]            1.4423      0.009    158.178      0.000       1.424       1.460
cap_color_l[T.True]            0.7960      0.010     80.774      0.000       0.777       0.815
cap_color_n[T.True]            1.1194      0.007    161.432      0.000       1.106       1.133
cap_color_o[T.True]            2.0711      0.008    267.799      0.000       2.056       2.086
cap_color_p[T.True]            2.2876      0.009    263.289      0.000       2.271       2.305
cap_color_r[T.True]            3.1605      0.010    321.860      0.000       3.141       3.180
cap_color_u[T.True]            1.8144      0.008    216.648      0.000       1.798       1.831
cap_color_w[T.True]            1.0440      0.007    144.165      0.000       1.030       1.058
cap_color_y[T.True]            1.4377      0.007    201.550      0.000       1.424       1.452
does_bruise_or_bleed_t[T.True]     0.0984      0.002     40.838      0.000       0.094       0.103
has_ring_t[T.True]             0.4962      0.002    223.373      0.000       0.492       0.501
habitat_g[T.True]              0.4812      0.003    173.356      0.000       0.476       0.487
habitat_h[T.True]              0.3263      0.005     67.603      0.000       0.317       0.336
habitat_l[T.True]             -0.3202      0.004    -80.715      0.000      -0.328      -0.312
habitat_m[T.True]             -0.4395      0.004   -103.631      0.000      -0.448      -0.431
habitat_p[T.True]             18.2376     63.609      0.287      0.774    -106.434     142.909
habitat_u[T.True]            -20.5452    319.552     -0.064      0.949    -646.856     605.766
habitat_w[T.True]            -20.7921    140.689     -0.148      0.883    -296.538     254.953
season_s[T.True]              -1.0029      0.004   -223.404      0.000      -1.012      -0.994
season_u[T.True]              -0.0872      0.002    -47.413      0.000      -0.091      -0.084
season_w[T.True]              -0.8362      0.003   -255.309      0.000      -0.843      -0.830
stem_height                   -0.0325      0.000    -96.864      0.000      -0.033      -0.032
stem_width                    -0.0308      0.000   -258.267      0.000      -0.031      -0.031
==============================================================================================
```

4. Metodologia e Resultados

A solução proposta envolve a aplicação de diferentes algoritmos de machine learning supervisionado, como Árvore de Decisão, Random Forest, e K-Nearest Neighbors (K-NN), para construir modelos preditivos baseados nas características categóricas do dataset. Serão utilizados métodos de pré-processamento, como a codificação de variáveis categóricas e a análise de correlação entre essas variáveis, para garantir que o modelo consiga aprender padrões significativos e realizar predições precisas.

A avaliação dos modelos será feita utilizando técnicas métricas de desempenho como acurácia, precisão, recall e F1-score, para garantir que o modelo tenha um desempenho robusto e confiável, minimizando tanto falsos positivos quanto falsos negativos.

5. Referências e Bibliografia

WANDO, Bwando. “Mushroom Overload”. Disponível em: https://www.kaggle.com/datasets/bwandowando/mushroom-overload. Acesso em: 27 de agosto de 2024.

O link acima faz referência a origem do dataset utilizado no projeto, com os dados em sua origem.

WAGNER, Dennis, Dominik H., Georges H, Patrick H. “Secondary Data”. Disponível em: https://github.com/ghattab/secondarydata/ Acesso em: 28 de agosto de 2024.

O conjunto de dados primário contém descrições de 173 espécies de cogumelos como entradas. Ele pode ser usado para simular cogumelos hipotéticos. O conjunto de dados secundário é um produto de tal simulação e contém 61.069 cogumelos hipotéticos. Ele pode ser usado para classificação binária.