Classificação Binária de Cogumelos Comestíveis e Venenosos

Victor G. Lima

Universidade Federal de Ouro Preto

07 de Julho de 2025

https://github.com/Victorgonl/PoisonousMushrooms

Sumário

- Introdução
- Trabalhos Relacionados
- Método
- Conclusão

Motivação

- Cogumelos são amplamentes utilizados na culinária.
- Cogumelos venenosos podem ser extremamente prejudiciais à saúde
- Existem diversas espécies com características similares.

Motivação

1 Introdução

Figura: Dois cogumelos: comestível *Agaricus campestris* (esquerda) e venenoso *Amanita phalloides* (direita).

Objetivo

- Classificar se um cogumelo é comestível ou vanenoso
- Características morfológicas.
- Também habitat e estações do ano.

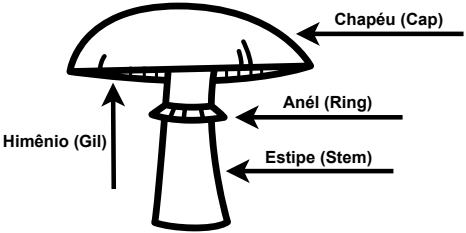

Figura: Algumas Características de um Cogumelo

Sumário

2 Trabalhos Relacionados

- Introdução
- Trabalhos Relacionados
- Método
- Conclusão

Dataset

- Novo dataset [Wagner et al., 2021] baseado em um outro mais antigo (1987).
- Expande o número de espécies (173) e famílias (23).

Dataset

2 Trabalhos Relacionados

Atributo	Tipo	Valores
class	Binário	poisonous=p, edible=e
cap-diameter	Contínuo	Número real em cm
cap-shape	Nominal	bell=b, conical=c, convex=x, flat=f, sunken=s, spherical=p, others=o
cap-surface	Nominal	fibrous=i, grooves=g, scaly=y, smooth=s, shiny=h, leathery=l, silky=k, sticky=t, wrinkled=w, fleshy=e
cap-color	Nominal	brown=n, buff=b, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y, blue=l, orange=o, black=k
does-bruise-bleed	Nominal	bruises-or-bleeding=t, no=f
gill-attachment	Nominal	adnate=a, adnexed=x, decurrent=d, free=e, sinuate=s, pores=p, none=f, unknown=?
gill-spacing	Nominal	close=c, distant=d, none=f
gill-color	Nominal	Mesmo que cap-color + none=f
stem-height	Contínuo	Número real em cm
stem-width	Contínuo	Número real em mm
stem-root	Nominal	bulbous=b, swollen=s, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r
stem-surface	Nominal	Mesmo que cap-surface + none=f
stem-color	Nominal	Mesmo que cap-color + none=f
veil-type	Nominal	partial=p, universal=u
veil-color	Nominal	Mesmo que cap-color + none=f
has-ring	Binário	ring=t, none=f
ring-type	Nominal	cobwebby=c, evanescent=e, flaring=r, grooved=g, large=l, pendant=p, sheathing=s, zone=z, scaly=y, movable=m, none=f, unknown=?
spore-print-color	Nominal	Mesmo que cap-color
habitat	Nominal	grasses=g, leaves=l, meadows=m, paths=p, heaths=h, urban=u, waste=w, woods=d
season	Nominal	spring=s, summer=u, autumn=a, winter=w

Figura: *Features* do dataset *Secondary Mushroom*.

- Em Tutuncu et al. [2022] e Tutuncu et al. [2025] são explorados algoritmos de aprendizado de máquina para classificação binária utilizando o *Secondary Mushroom*.

Algoritmo	Accuracy	Recall	Precision	F-Score
Naive Bayes	90.99	98.60	86.04	91.89
Decision Tree	98.82	97.72	100.00	98.85
SVM	99.98	100.00	99.95	99.98
AdaBoost	100.00	100.00	100.00	100.00

Figura: Resultados de diferentes algoritmos de aprendizado de máquina com o *Secondary Mushroom*.

Sumário

3 Método

- Introdução
- Trabalhos Relacionados
- Método
- Conclusão

Dataset do Kaggle

- *Mushroom Dataset* do desafio *Binary Prediction of Poisonous Mushrooms*[a].

Partição	Instâncias	Porcentagem	Comestíveis	Venenosos
Treino	3.116.945	60%	1.411.549 (45%)	1.705.396 (55%)
Teste	2.077.964	40%	?	?

Figura: Informações sobre o dataset de cogumelos do Kaggle.

[a] https://www.kaggle.com/competitions/playground-series-s4e8

Etapas

- Pré-processamento do dataset.
- Mineração de padrões frequentes.
- Avaliação experimental.

Etapas da Avaliação Experimental

- Escolha do modelo
- Métrica de avaliação
- Escopo para busca de melhor combinação de hiper-parâmetros.
- Experimento em *hold-oute 10-fold cross-validation*.
- Geração de modelos para solucionar problema no Kaggle.

Modelo Utilizado

- CatBoost Prokhorenkova et al. [2018].
- Consegue lidar bem com diferentes *features* nominais.
- Tem suporte para conjuntos grandes sem ocorrência de *overfitting*.
- Permite aceleração por meio de GPU.

Métrica

3 Método

- Foi utilizado como principal a *Matthews Correlation Coefficient* (MCC).

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

  - 1 indica uma predição perfeita;
  - 0 indica uma predição aleatória;
  - -1 indica uma total discordância entre predições e valores reais.
- Também avaliado a acurácia, precisão, revocação e F-Score.

Particionamento do dataset

- O dataset (treino do Kaggle) foi divido em 10 partes iguais.
- Para *hold-out*: 1-7 (treino), 8 (validação) e 9-10 (teste).
- Conjunto de validação utilizado para busca de melhor combinação de hiper-parâmetros.

Busca por hiper-parâmetros

3 Método

- Realizada 100 tentativas no experimento *hold-out*.
- Realizada 25 tentativas para cada iteração do *10-fold-cross validation*.

Busca por hiper-parâmetros

3 Método

```python
def create_objective(X_train, y_train, X_val, y_val, device="CPU"):
    def objective(trial: optuna.Trial):
        params = {
            "iterations": trial.suggest_categorical("iterations", [250, 500, 750, 1000]),
            "learning_rate": trial.suggest_float("learning_rate", 0.001, 0.1, log=True),
            "depth": int(trial.suggest_int("depth", low=4, high=12)),
            "l2_leaf_reg": int(trial.suggest_int("l2_leaf_reg", low=2, high=10)),
            "border_count": trial.suggest_categorical("border_count", [64, 128, 254]),
            "loss_function": "Logloss",
            "feature_border_type": "GreedyLogSum",
            "early_stopping_rounds": 50,
        }

        model = CatBoostClassifier(verbose=0, use_best_model=True, task_type=device, **params)

        model.fit(X_train, y_train, eval_set=(X_val, y_val), verbose=False)
        y_pred = model.predict(X_val)
        mcc = matthews_corrcoef(y_val, y_pred)
        return mcc

    return objective
```

Figura: Função objetivo implementada para busca de melhores hiper-parâmetros.

Experimento de *hold-out*

- Melhor combinação encontrada na tentativa 76 com $MCC = 0.98199$.

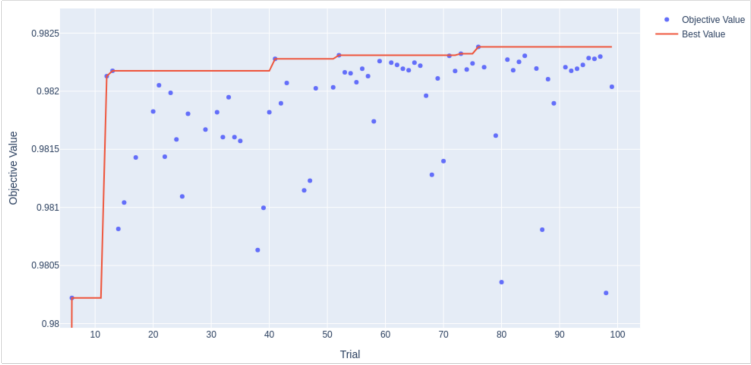

Figura: Evolução na busca de melhor combinação de hiper-parâmetros no *hold-out*.

Experimento de *hold-out*

3 Método

		Hiper-Parâmetro				
Tentativa	**MCC**	`iterations`	`learning_rate`	`depth`	`l2_leaf_reg`	`border_count`
76	0.98238	1000	0.07320	12	7	64

Figura: Melhor combinação encontrada no experimento de *hold-out*.

Accuracy	**Macro Precision**	**Macro Recall**	**Macro F-Score**	**MCC**
0.99107	0.99091	0.99109	0.99010	0.98199

Figura: Resultados no experimento de *hold-out*.

Experimento de *10-fold cross-validation*

3 Método

Algorithm 1 Validação Cruzada em K partes com Busca de Hiper-Parâmetros.

Seja o dataset dividido em K partições.
for $i = 1$ **to** K **do**:
    `teste` $\leftarrow k_i$.
    `validação` $\leftarrow k_{(i+1 \bmod K)}$.
    `treino` $\leftarrow$ restante das partições.
    Busca melhor combinação de hiper-parâmetros usando `treino` e `validação`.
    Avalia o melhor modelo encontrado com `teste` e salva as métricas.
end for
Calcula a média e o desvio-padrão das métricas salvas.

Figura: Algoritmo implementado para *k-fold cross-validation*.

Experimento de *10-fold cross-validation*

Fold	Accuracy	Precision	Recall	F1-score	MCC
1	0.99081	0.99212	0.99109	0.99160	0.98147
2	0.99085	0.99286	0.99039	0.99162	0.98155
3	0.99066	0.99232	0.99054	0.99143	0.98117
4	0.99115	0.99268	0.99117	0.99192	0.98213
5	0.99117	0.99292	0.99092	0.99192	0.98218
6	0.99077	0.99271	0.99041	0.99156	0.98138
7	0.99072	0.99241	0.99062	0.99151	0.98127
8	0.99095	0.99269	0.99076	0.99173	0.98173
9	0.99109	0.99282	0.99086	0.99184	0.98203
10	0.99108	0.99275	0.99094	0.99184	0.98200
$\bar{x} \pm \sigma$	0.99092 ± 0.00018	0.99263 ± 0.00025	0.99077 ± 0.00026	0.99170 ± 0.00017	0.98169 ± 0.00036

Figura: Resultados no experimento de *10-fold cross-validation*.

- Um modelo treinado com todos os dados de treino utilizando a melhor combinação de hiper-parâmetros encontrada no experimento de *hold-out*.
- Um *Voting Classifier* resultante da combinação dos 10 modelos gerados pelo *10-fold cross-validation*.
  - Hard
  - Soft

Submission and Description		Private Score	Public Score
submission.csv Complete (after deadline)	CatBoostClassifier from hold-out with best hp using all train data.	0.98201	0.98219
solution_voting_hard.csv Complete (after deadline)	Hard VotingClassifier using 10 CatBoostClassifier from 10-fold-cv with best hp.	0.98202	0.98238
solution_voting_soft.csv Complete (after deadline)	Soft VotingClassifier using 10 CatBoostClassifier from 10-fold-cv with best hp.	0.98206	0.98237

Figura: Resultados obtidos após submissão no Kaggle.

Kaggle

3 Método

#	△	Team	Members	Score	Entries	Last	Solution
1	—	Optimistix		0.98514	124	10mo	
2	▲ 194	Sujal Neupane		0.98511	25	10mo	
3	▲ 7	AutoML Grandmasters		0.98510	6	1y	
4	▲ 144	Elephant In The Shroom 🍄		0.98509	119	10mo	
5	▲ 93	nathankim0917		0.98508	7	10mo	
6	▲ 5	Tilii		0.98508	73	10mo	
7	▲ 16	GMLB_korea_mapo_03		0.98508	47	10mo	
8	▲ 231	Jack Lee		0.98507	5	10mo	
9	▲ 36	GMLB_Fire Coral		0.98507	40	10mo	
10	▲ 130	Mahdi Ravaghi		0.98506	127	10mo	

Figura: Top 10 no ranking do desafio no Kaggle.

Sumário

4 Conclusão

- Introdução
- Trabalhos Relacionados
- Método
- Conclusão

Conclusão

- Foi abordado a classificação de cogumelos entre comestíveis e venenosos aplicando técnicas de Mineração de Dados.
- O método utilizado avaliou a solução nas modalidades *hold-out* e *10-fold cross-validation*.
- A solução obteve uma correlação forte, avaliada pela métrica MCC junto ao conjunto de teste.

Referências

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6639–6649, Red Hook, NY, USA, 2018. Curran Associates Inc.

Kemal Tutuncu, Ilkay Cinar, Ramazan Kursun, and Murat Koklu. Edible and poisonous mushrooms classification by machine learning algorithms. In *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4, 2022.

Kemal Tutuncu, Ilkay Cinar, Ramazan Kursun, and Murat Koklu. Research on mushroom classification based on xgb technology. In *2025 International Conference on Biomaterials, Computer Engineering and Ecological Environment*, volume 13, 2025.

Dennis Wagner, Dominik Heider, and Georges Hattab. Mushroom data creation, curation, and simulation to support classification tasks. *Scientific Reports*, 11, 2021.