

Artificial Intelligence (AI) CS204A

MUSHROOM CLASSIFICATION SUMMATIVE ASSESSMENT



Photo by Pixabay: <https://www.pexels.com/photo/food-wood-kitchen-cutting-board-36438/>, Photo by Egor Kamelev: <https://www.pexels.com/photo/closeup-photo-of-red-and-white-mushroom-757292/>

[My Portal – Blackboard Learn \(up.education\)](#)

Outline

[Outline](#)

[1.Scenario](#)

[2.Experimental Evaluation](#)

[3. Results](#)

[Attribute Information:](#)

[4. Discussion & Conclusion](#)

[References](#)

1.Scenario

Using a Decision Tree Classifier; the aim of this assessment is to find any correlation between the mushroom species' features and if it is edible or not. It is also equally important to test the accuracy and performance of this model. The Mushroom dataset [1] is sourced from the UCI ML (University of California Irvine Machine Learning) Repository and includes descriptions of over 8,100 hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility/not recommended. This latter class was combined with the poisonous one [2].

2.Experimental Evaluation

Manual Evaluation Process (hypothetical discussion)

Without the use of machine learning techniques, the overall process would be very tedious and would consume a lot of time and energy. Tasks such as categorising attributes of each mushroom image (features such as odor, cap-shape, cap-colour etc) require attention to detail on a large scale; it would be almost unavoidable to encounter human error in the process of analysing the data. There is also higher risk of bias classification; if each image was manually scanned by human eyes, care must be taken to avoid being in the wrong state e.g. fatigued, limited judgement or being (un-awaringly) impaired (e.g. colour blindness or blurred vision). Unless the person/people evaluating the data had infinite time on their hands, it is highly unlikely there would be time to test more than one model, let alone compare results to another model.

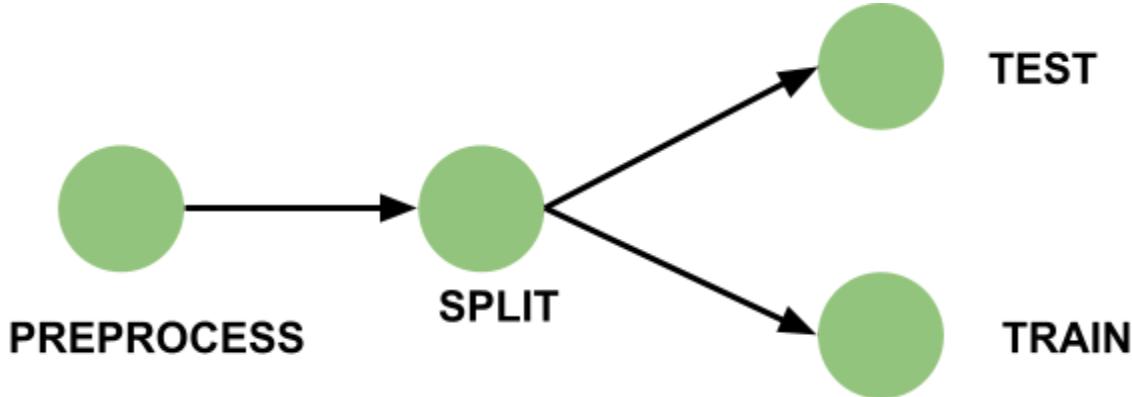
Improvements with Machine Learning

Using machine learning models to examine big sets of data allows us to apply more than one process, often at the same time. A possible advantage of this is when a ML model finds a correlation between some features that we could not spot during the analysis stage. Algorithms compute at such speeds that it makes the task of processing vast amounts of data much easier and efficient.

If we are not sure if a feature is relative or not then it would be better to include it anyway, so that we don't risk losing helpful information. We can always remove the ones we believe are irrelevant later and try the model again; then compare the results and decide which model to use. A final decision on the best model is often based on their accuracy, although other factors such as speed and computing power can also influence a decision on the best approach.

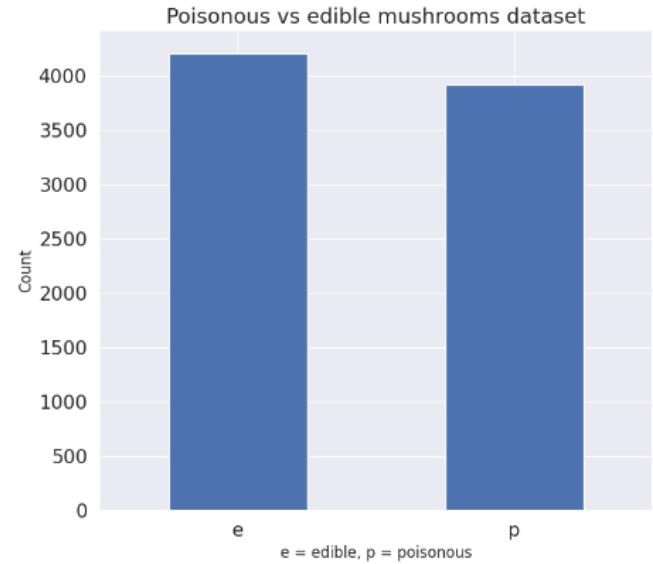
Data Analysis

The diagram below is the model pipeline that has been used for the purpose of this assessment. The pipeline is made of 4 tasks (green circles):



Preprocessing the Data

Raw data (in this case I used a csv file), and performed the following steps: examine the dataset shape, identify the number of features and classes and count the frequency of the top features that occurred across the sample images. It was seen that there were two classes; poisonous and edible and when visualised as a bar plot (*right image*) it was easy to notice the count for each of these classes were fairly close (edible: 4208, poisonous: 3916). This is described as being *balanced*.



Another part of the preprocessing stage is data transformation, which is when formatting changes are made to the original data in order to organise it better. The original dataset used letters to categorise the values for each feature (*see Attribute Information table*) and this was then encoded to numerical format which is a necessary step in order to effectively fit and evaluate the classifier model later on. One of the 22 features, `veil-type` was also removed as it was not contributing to the data after the previous change.

Splitting, Testing and Training the Data

After the preprocessing stage, the data is then split into two sets; namely training set and testing set to avoid overfitting. Overfitting is when a model learns to use an overly specific function which performs excellently on one set of data but does not perform well when the model takes in new sets of data it has not seen before. It is like the model creates a bias from that first dataset which affects its performance when new data is fed into it.

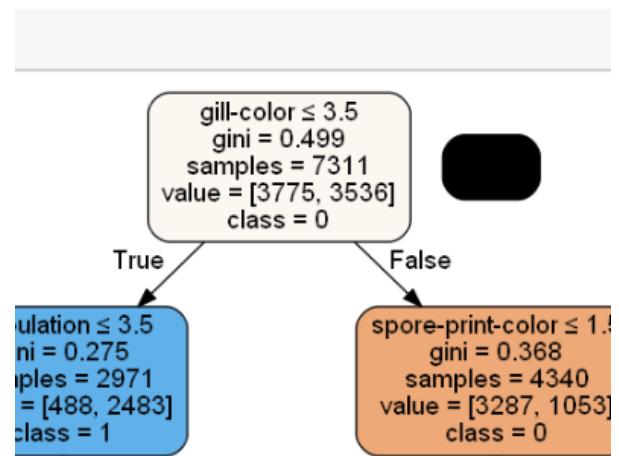
In this case, 90% of the original dataset was allocated as the training set and 10% was allocated for the testing set. A third set which was not used in this process is the validation set, which gives a sense of how well the model performs on images that are not being used in training.

3. Results

This section discusses the key visualisations used to display the model's results.

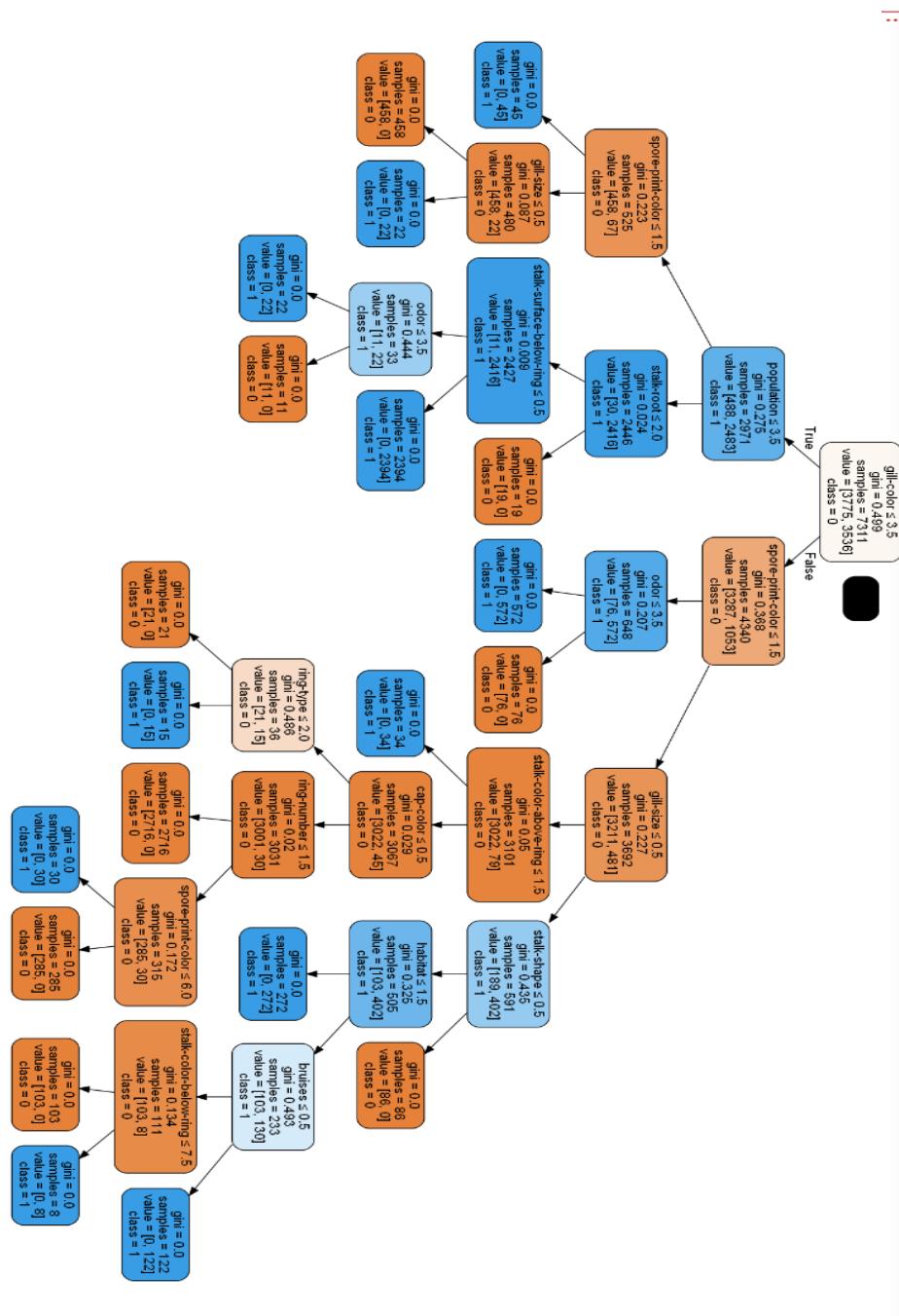
The Decision Tree

Decision trees start with the entire dataset and inspect every feature by reading all the possible values of a feature and then it decides which value would best split the dataset into two regions (a true side and false side). It is made up of the root node (*image on the right*), decision or internal nodes and then finally leaf nodes (the boxes that do not split any further). The blue nodes are when the



poisonous class is true and the orange ones are when it is false (edible class).

There are two main types of trees, decision tree regressors and decision tree classifiers; the latter is used for this assessment. Decision tree classifiers rely on two metrics: gini (impurity) and information gain. Gini scores how ‘pure’ the splits are from each node for example a node that has class=0, gini=0.499 is equal to saying that the edible class is 49% true for this node.



Attribute Information:

- 2 classes: edible=e, poisonous=p,
- 22 attributes/features (listed below)

No.	ATTRIBUTE	ATTRIBUTE CATEGORY (ranging between 2-12 values per attribute)											
1	cap-shape:	bell=b,	conical=c,	convex=x,	flat=f,	knobbed=k,	sunken=s						
2	cap-surface:	fibrous=f,	grooves=g,	scaly=y,	smooth=s								
3	cap-color:	brown=n,	buff=b,	cinnamon=c,	gray=g,	green=r,	pink=p,	purple=u,	red=e,	white=w,	yellow=y		
4	bruises:	bruises=t,	no=f										
5	odor:	almond=a	anise=l,	creosote=c,	fishy=y,	foul=f,	musty=m,	none=n,	pungent=p,	spicy=s			
6	gill-attachmen t:	attached=a,	descendin	free=f,	notched=n								
7	gill-spacing:	close=c,	crowded=w,	distant=d									
8	gill-size:	broad=b,	narrow=n										
9	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
10	stalk-shape:	enlarging=e,	tapering=t										
11	stalk-root:	bulbous=b,	club=c,	cup=u,	equal=e,	rhizomorp	rooted=r,	missing=?					
12	stalk-surface- above-ring:	fibrous=f,	scaly=y,	silky=k,	smooth=s								

13	stalk-surface-below-ring:	fibrous=f,	scaly=y,	silky=k,	smooth=s								
14	stalk-color-above-ring:	brown=n,	buff=b,	cinnamon=c,	gray=g,	orange=o,	pink=p,	red=e,	white=w,	yellow=y			
15	stalk-color-below-ring:	brown=n,	buff=b,	cinnamon=c,	gray=g,	orange=o,	pink=p,	red=e,	white=w,	yellow=y			
16	veil-type:	partial=p,	universal=u										
17	veil-color:	brown=n,	orange=o,	white=w,	yellow=y								
18	ring-number:	none=n,	one=o,two=t										
19	ring-type:	cobwebby=c,	evanescent=e,	flaring=f,	large=l,	none=n,	pendant=p,	sheathing=s,	zone=z				
20	spore-print-color:	black=k,	brown=n,	buff=b,	chocolate=h,	green=r,	orange=o,	purple=u,	white=w,	yellow=y			
21	population:	abundant=a,	clustered=c,	numerous=n,	scattered=s,	several=v,	solitary=y						
22	habitat:	grasses=g,	leaves=l,	meadows=m	paths=p,	urban=u,	waste=w,	woods=d					

Precision and Recall Scores

Precision and Recall scores are used to evaluate the effectiveness of a model; precision is a measure of quality and recall is a measure of quantity. In this case we have 100% scores for both, which correlates with the accuracy score (also 100%).

Decision Tree Classifier report:

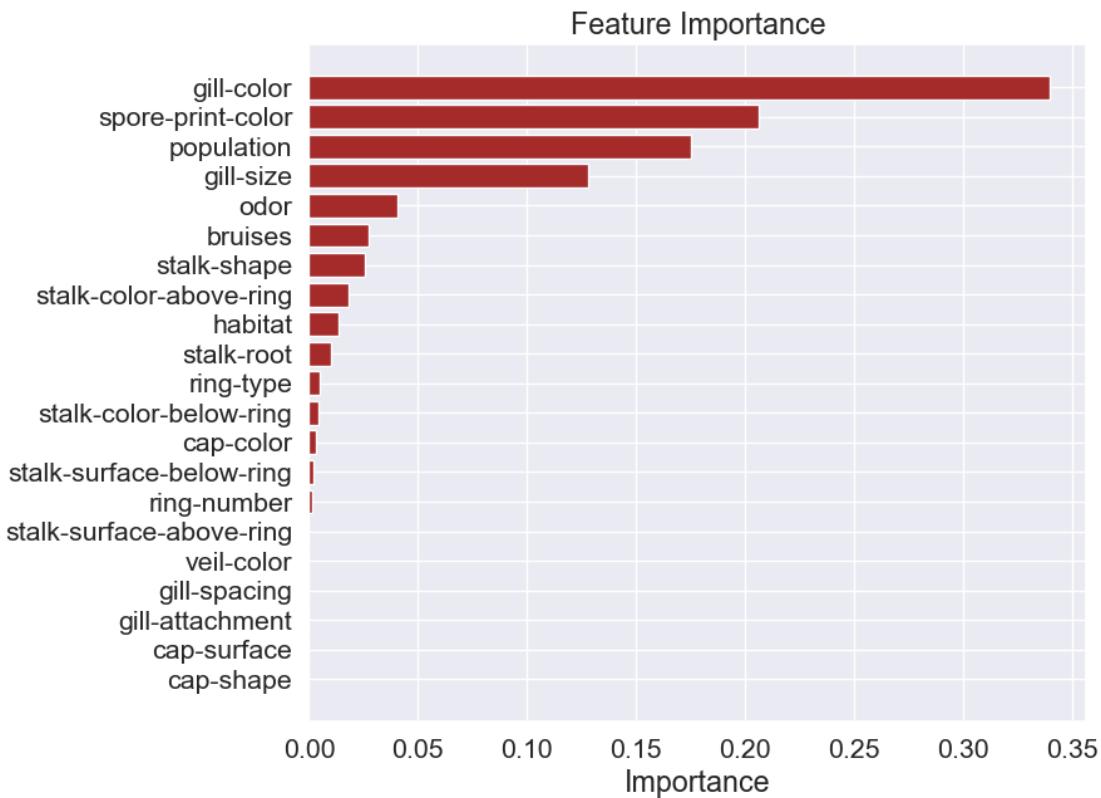
	precision	recall	f1-score	support
0	1.00	1.00	1.00	433
1	1.00	1.00	1.00	380
accuracy			1.00	813
macro avg	1.00	1.00	1.00	813
weighted avg	1.00	1.00	1.00	813

```
In [34]: #print test accuracy
print("Test Accuracy: {}".format(round(dt.score(X_test, y_test)*100, 2)))
```

Test Accuracy: 100.0%

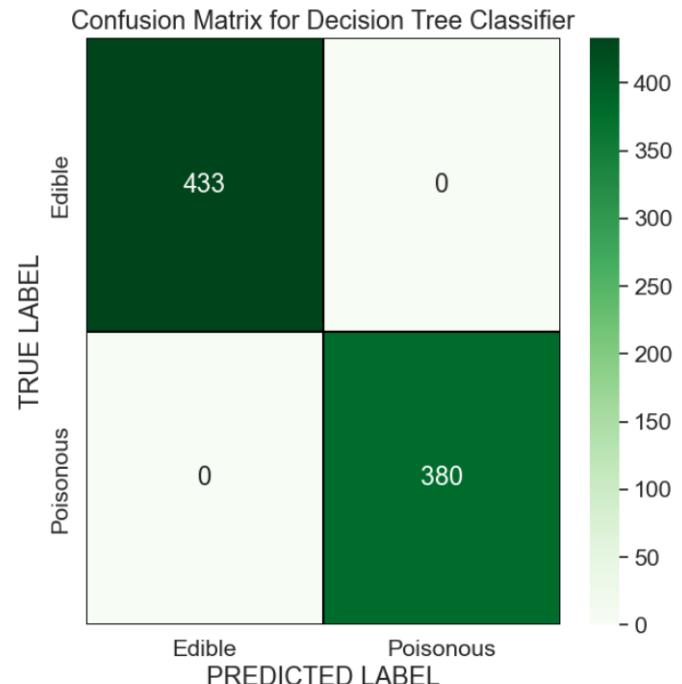
About Feature Importance

The features or attributes are the individual properties that are fed into the model (in this case, the Decision Tree). Feature Importance calculates a score for each feature that determines how relevant that feature contributed to the model's prediction method. By discovering and removing the irrelevant features (features with lowest/no score), it is then possible to increase the model's speed or improve its performance. In this case, `gill-colour` was by far the most relevant, with `spore-print-color`, `population` and `gill-size` following next, scoring quite close together.



About Confusion Matrix

Confusion Matrices are used in conjunction with accuracy scores to summarise the performance of a classification algorithm. They calculate how many times the model predicts a certain class (PREDICTED LABEL) against how many times the model got those predictions correct or not (TRUE LABEL). Because both the accuracy score and test accuracy scores for this model were 100%, there were zero instances where mushroom samples were predicted to be edible but turned out to be poisonous and vice versa.



4. Discussion & Conclusion

This section discusses how the model could be improved in terms of better performance and a reflection on the current limitations which relates to future work in this field.

100% Accuracy?

It is quite unusual to get such high accuracy on the first attempt without having to finetune parameters. Although it is possible this may be an exception, it is worth exploring other methods to check for potential overfitting.

The splitting of the data was an aspect that needed more experimentation. I suspect there was an element of overfitting with the model's performance so I think including a validation set of 20% and reducing the training set to 70% would reduce that risk further.

The Mushroom Dataset

The chosen dataset only focuses on two family species of mushroom and was donated to UCI ML in 1987, which is quite a long time ago. The primary reason this dataset was selected was due to lack of time and resources to create my own (also wrong season in NZ) and this dataset is currently the most popular for classification purposes. I suspect there is a huge gap in data for fungi species in general and I do think it would be a hefty but worthwhile task to create more datasets relating to this field.

References

[1] <https://www.kaggle.com/datasets/uciml/mushroom-classification>

[2] <https://archive.ics.uci.edu/ml/datasets/Mushroom>

Feature Importance: <https://www.baeldung.com/cs/ml-feature-importance>

Confusion Matrix: <https://machinelearningmastery.com/confusion-matrix-machine-learning/>

Splitting: <https://blog.roboflow.com/train-test-split/>

Decision trees: <https://medium.com/analytics-vidhya/decision-trees-for-dummies-a8e3c00c5e2e>

100% Accuracy? <https://www.kaggle.com/datasets/uciml/mushroom-classification/discussion/198420>