Module 2: Predicting whether a Mushroom is edible or poisonous using Decision Tree - Entropy

Snehal Sham Tikone

Northeastern University

Course Number: ALY 6040 Data Mining Applications

Prof. Dr. Shanu Sushmita

April 24, 2022

Introduction

Mushroom dataset contained below columns:

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   class                     8124 non-null   object
 1   cap-shape                 8124 non-null   object
 2   cap-surface               8124 non-null   object
 3   cap-color                 8124 non-null   object
 4   bruises                   8124 non-null   object
 5   odor                      8124 non-null   object
 6   gill-attachment           8124 non-null   object
 7   gill-spacing              8124 non-null   object
 8   gill-size                 8124 non-null   object
 9   gill-color                8124 non-null   object
 10  stalk-shape               8124 non-null   object
 11  stalk-root                8124 non-null   object
 12  stalk-surface-above-ring  8124 non-null   object
 13  stalk-surface-below-ring  8124 non-null   object
 14  stalk-color-above-ring    8124 non-null   object
 15  stalk-color-below-ring    8124 non-null   object
 16  veil-type                 8124 non-null   object
 17  veil-color                8124 non-null   object
 18  ring-number               8124 non-null   object
 19  ring-type                 8124 non-null   object
 20  spore-print-color         8124 non-null   object
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
memory usage: 1.4+ MB
None
```

Sample view of top 10 records in the dataset:

Out[55]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	s
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	n
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	n
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	s
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	a
5	e	x	y	y	t	a	f	c	b	n	...	s	w	w	p	w	o	p	k	n
6	e	b	s	w	t	a	f	c	b	g	...	s	w	w	p	w	o	p	k	n
7	e	b	y	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	s
8	p	x	y	w	t	p	f	c	n	p	...	s	w	w	p	w	o	p	k	v
9	e	b	s	y	t	a	f	c	b	g	...	s	w	w	p	w	o	p	k	s

Data Description:

Out[8]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	popul
count	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124	...	8124	8124	8124	8124	8124	8124	8124	8124	
unique	2	6	4	10	2	9	2	2	2	12	...	4	9	9	1	4	3	5	9	
top	e	x	y	n	f	n	f	c	b	b	...	s	w	w	p	w	o	p	w	
freq	4208	3656	3244	2284	4748	3528	7914	6812	5612	1728	...	4936	4464	4384	8124	7924	7488	3968	2388	

4 rows × 23 columns

All the variables in the dataset are objects:

```
Out[58]: class                       object
         cap-shape                   object
         cap-surface                 object
         cap-color                   object
         bruises                     object
         odor                        object
         gill-attachment             object
         gill-spacing                object
         gill-size                   object
         gill-color                  object
         stalk-shape                 object
         stalk-root                  object
         stalk-surface-above-ring    object
         stalk-surface-below-ring    object
         stalk-color-above-ring      object
         stalk-color-below-ring      object
         veil-type                   object
         veil-color                  object
         ring-number                 object
         ring-type                   object
         spore-print-color           object
         population                  object
         habitat                     object
         dtype: object
```

Null Data:

```
Null Data:
 class                          0
cap-shape                       0
cap-surface                     0
cap-color                       0
bruises                         0
odor                            0
gill-attachment                 0
gill-spacing                    0
gill-size                       0
gill-color                      0
stalk-shape                     0
stalk-root                      0
stalk-surface-above-ring        0
stalk-surface-below-ring        0
stalk-color-above-ring          0
stalk-color-below-ring          0
veil-type                       0
veil-color                      0
ring-number                     0
ring-type                       0
spore-print-color               0
population                      0
habitat                         0
dtype: int64
```

Summary of Data Cleaning and Analysis:

1. There are 23 columns and 8124 rows in the dataset
2. All the variables are categorical.
3. There are no null values in the dataset.
4. No duplicate rows.
5. The class is the target variable while other are the features.
6. The class can have either values p - mushroom is poisonous e - mushroom is edible

Decision Tree:

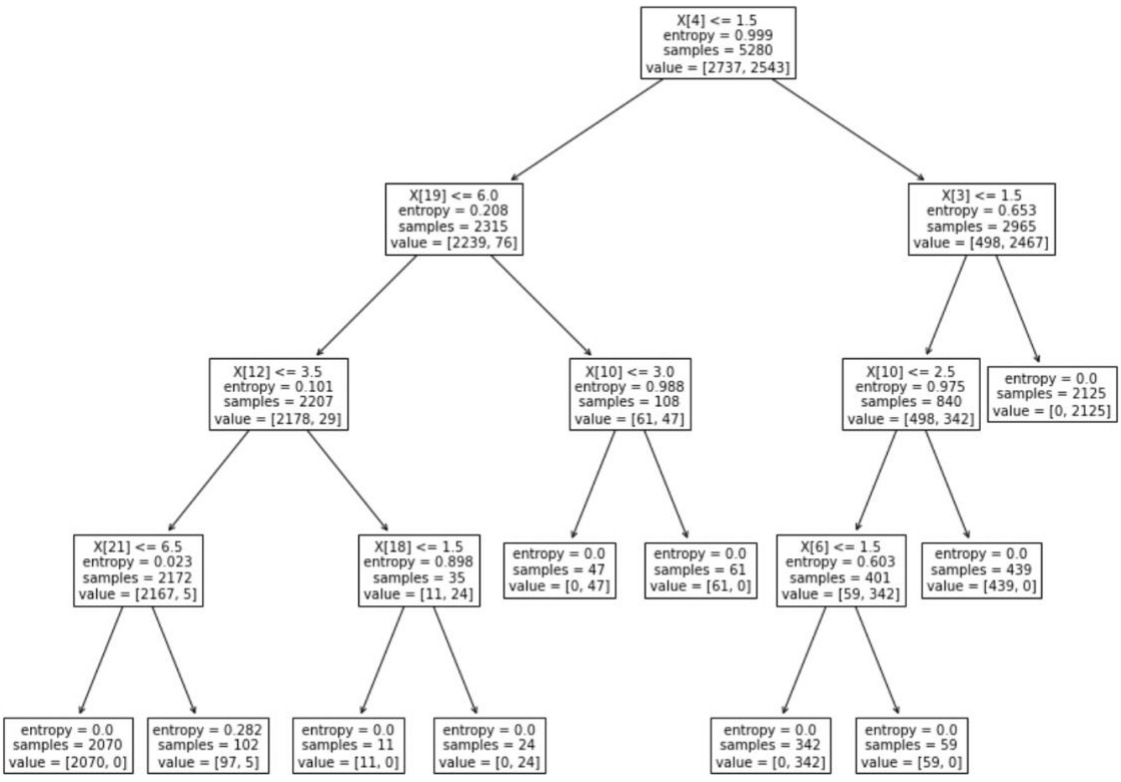
Divided the dataset in 65:35 ratio of Train : Test.

Applied Decision Tree – criteria entropy and predicted whether the mushroom is edible or not.

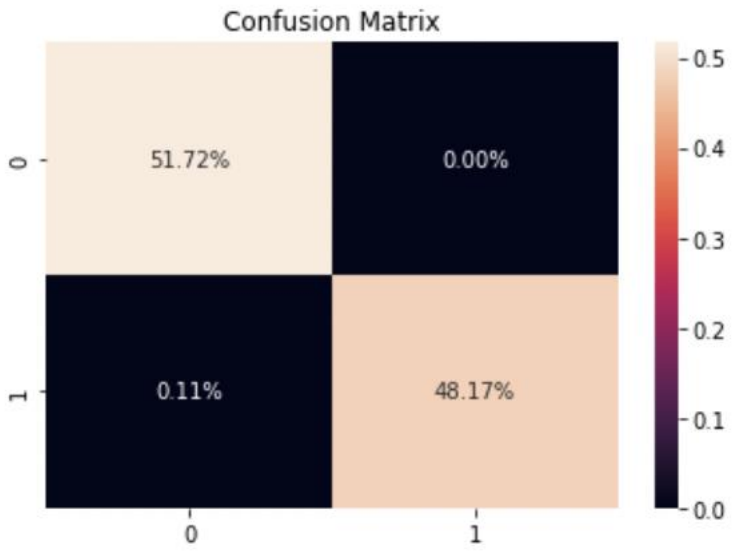
Allowed up to 4 levels of depth for the tree.

The accuracy score of the prediction is very high:

```
Training set score: 0.9991
Test set score: 0.9989
```



Confusion Matrix:



True Positive: 51.72% times the model predicted the positive class as positive

False Positive: 0% time the model incorrectly predicted the negative class as positive

False Negative: 0.11 % which is very negligible the model predicted the positive class as negative

True Negative: 48.17% times the model predicted the negative class as negative.

Conclusion: Based on the accuracy of the decision tree model 99.89%, with depth level as 4, the model has proved to most accurate. Also, the data was already clean with no duplicates and the confusion matrix also shows that the true positive and the true negative parts of the matrix have higher rates.

References

Abbas, M. M. (2021, November 26). *Count Unique Values in NumPy Array*. Delft Stack. https://www.delftstack.com/howto/numpy/python-numpy-value-counts/#:%7E:text=To%20count%20each%20unique%20element's,the%20array%20in%20ascending%20order.

Mohajon, J. (2021, December 14). *Confusion Matrix for Your Multi-Class Machine Learning Model*. Medium. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

pandas.DataFrame.plot.density — pandas 1.4.2 documentation. (n.d.). Pandas. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.density.html

Raj, A. (2022, January 4). *An Exhaustive Guide to Decision Tree Classification in Python 3.x*. Medium. https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f

T, D. (2021, December 11). *Confusion Matrix Visualization - Dennis T*. Medium. https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea