FACULTY OF COMPUTING
UTM Johor Bahru

SECB3203-01(PROGRAMMING FOR BIOINFORMATIC)

Semester 01 2025/2026
Section 01

Progress 2

Faculty of Computing

Dataset:
https://www.kaggle.com/datasets/uciml/mushroom-classification

STUDENT NAME	MATRIC NUMBER
NURUL NATALIA BINTI ROSNIZAM	A23CS0165
ADLINA NARISYA BINTI ISMAIL	A23CS0033
NAZATUL NADHIRAH BINTI SABTU	A23CS0144

2.0 Data Collection and Preprocessing

2.1 Importing Dataset

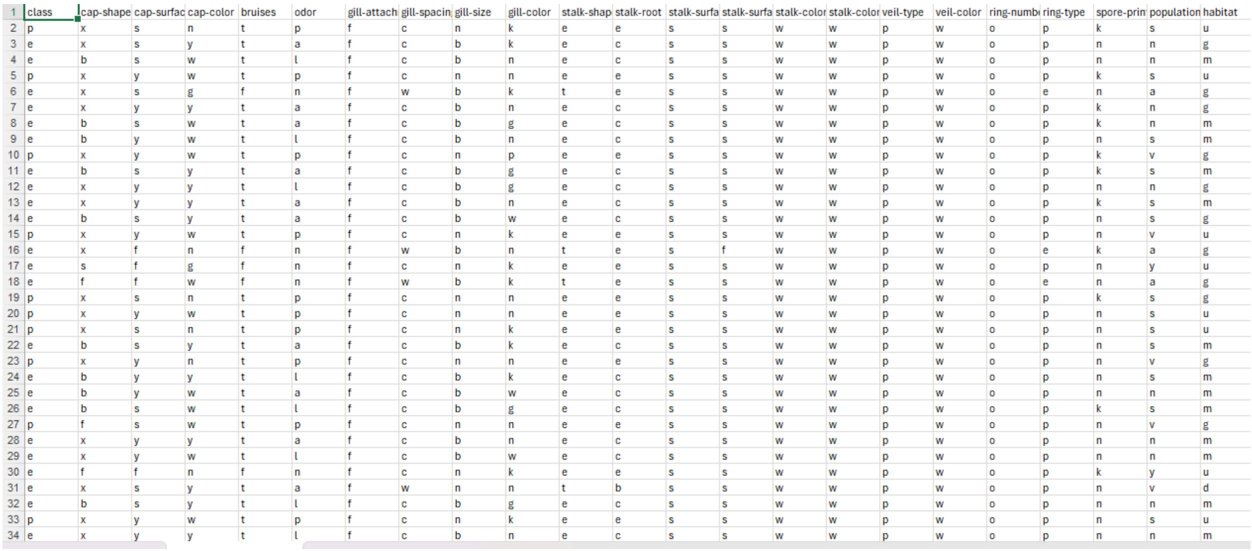
2.1.1 Understanding the Data

In this research, a mushroom classification dataset is utilized to predict whether a mushroom is edible or poisonous based on its physical characteristics. The dataset contains various categorical attributes describing mushroom properties such as class, cap shape, cap color, odor, gill size, stalk shape, habitat, and other morphological features. These attributes are critical in identifying patterns that distinguish edible mushrooms from poisonous ones.

The dataset was obtained from Kaggle and is widely used for classification and data mining research. It consists of 8124 mushroom samples (rows) with 24 features (columns), including one target attribute representing the class of the mushroom. Each feature plays an important role in understanding mushroom toxicity and edibility.

Our dataset from Kaggle: https://www.kaggle.com/code/rocklen/mushroom-classification/input

Figure 2.0: Raw dataset from kaggle



1	class	cap-shape	cap-surfac	cap-color	bruises	odor	gill-attach	gill-spacin	gill-size	gill-color	stalk-shap	stalk-root	stalk-surfa	stalk-surfa	stalk-color	stalk-color	veil-type	veil-color	ring-numb	ring-type	spore-prin	population	habitat
2	p	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	s	u
3	e	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	g
4	e	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m
5	p	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	u
6	e	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g
7	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g
8	e	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	n	m
9	e	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	m
10	p	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	w	o	p	k	v	g
11	e	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m
12	e	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	g
13	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	s	m
14	e	b	s	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	s	g
15	p	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	v	u
16	e	x	f	n	f	n	f	w	b	n	t	e	s	f	w	w	p	w	o	e	k	a	g
17	e	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	y	u
18	e	f	f	w	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g
19	p	x	s	n	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	g
20	p	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	s	u
21	p	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	s	u
22	e	b	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	s	m
23	p	x	y	n	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	v	g
24	e	b	y	y	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	s	m
25	e	b	y	w	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	n	m
26	e	b	s	w	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m
27	p	f	s	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	v	g
28	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m
29	e	x	y	w	t	l	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	n	m
30	e	f	f	n	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	y	u
31	e	x	s	y	t	a	f	w	n	n	t	b	s	s	w	w	p	w	o	p	n	v	d
32	e	b	s	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	m
33	p	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	s	u
34	e	x	y	y	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m

Figure 2.0 illustrates the mushroom dataset obtained from Kaggle, consisting of 8124 instances and 23 attributes.

Table 2.1 List of selected attributes mushroom dataset

Feature	Explanation
Class	Indicates mushroom edibility: **e** – edible, **p** – poisonous.
Cap-shape	Describes the shape of the mushroom cap: **b** – bell, **c** – conical, **x** – convex, **f** – flat, **k** – knobbed, **s** – sunken.
Cap-surface	Describes the surface texture of the cap: **f** – fibrous, **g** – grooved, **y** – scaly, **s** – smooth.
Population	Indicates how the mushroom population appears in nature: **c** – clustered **n** – numerous **s** – scattered **v** – several **y** – solitary **a** - abundant
Habitat	Describes the environment where the mushroom commonly grows: **g** – grasses **l** – leaves **m** – meadows **p** – paths **u** – urban areas **w** – waste areas

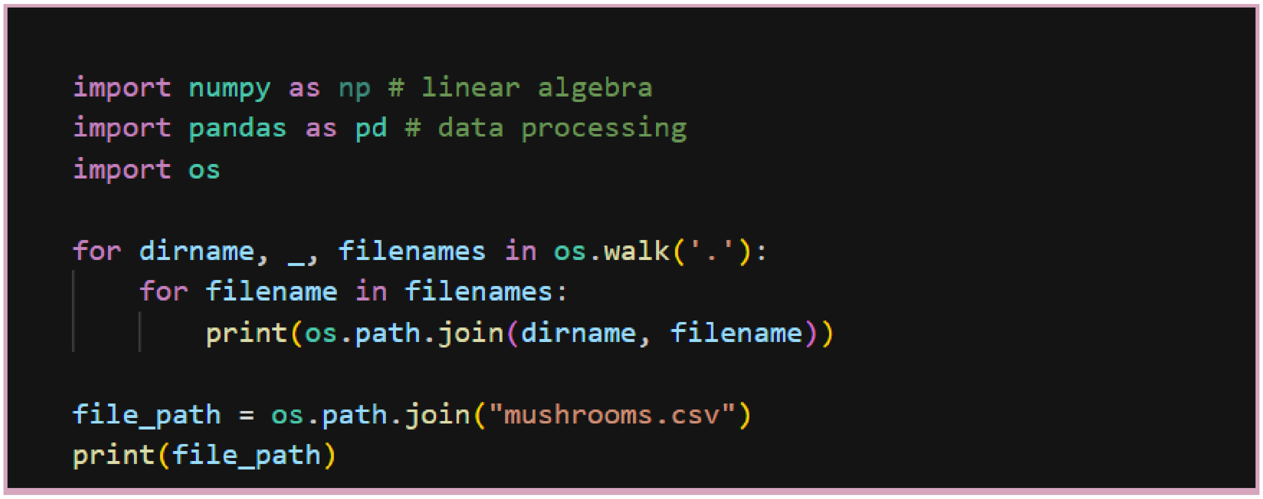
	d – woods
Spore-print-color	Indicates the color of the mushroom’s spore print: **k** – black **n** – brown **b** – buff **h** – chocolate **r** – green **o** – orange **u** – purple **w** – white **y** – yellow
Veil-color	Describes the color of the veil covering the mushroom when young: **n** – brown **o** – orange **w** – white **y** – yellow
Gill-attachment	Describes how the gills are attached to the stalk: **a** – attached **f** – free
Gill-size	Describes the size of the gills under the mushroom cap: **b** – broad **n** – narrow

In **Table 2.1,** show selected features along with their explanations and name that are relevant for mushroom classification.

2.1.2 Importing and Exporting Data in Python

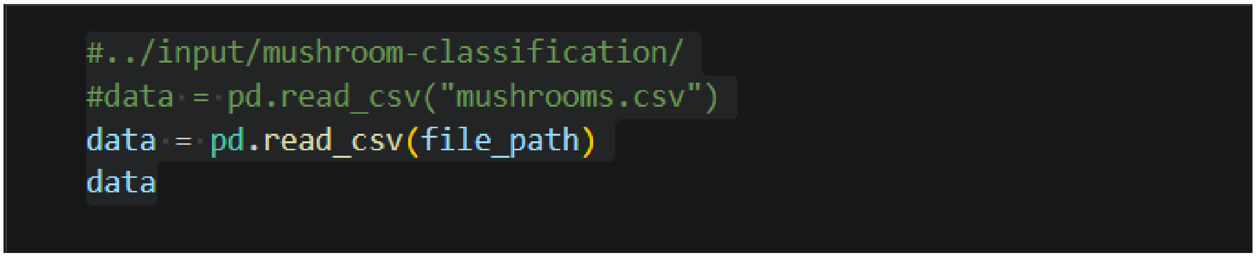
The dataset is provided in CSV format. We import it into Python using the numpy, pandas, os library:

Figure 2.1: Importing dataset using pandas library



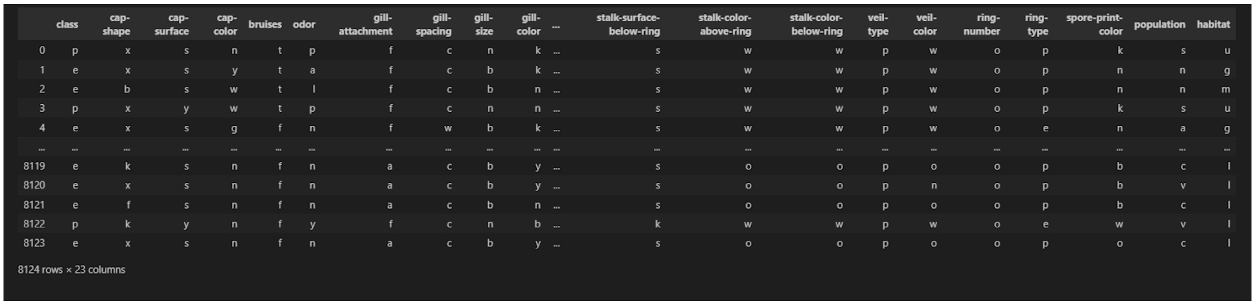
```python
import numpy as np # linear algebra
import pandas as pd # data processing
import os

for dirname, _, filenames in os.walk('.'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

file_path = os.path.join("mushrooms.csv")
print(file_path)
```

Figure 2.2: Read CSV file



```python
#../input/mushroom-classification/
#data = pd.read_csv("mushrooms.csv")
data = pd.read_csv(file_path)
data
```

Figure 2.3: Display the result of raw mushroom dataset



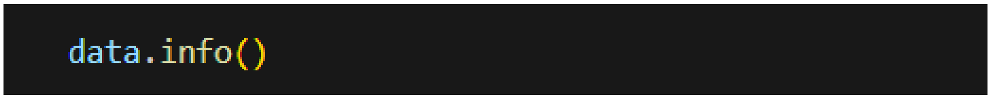
	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	s	u
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	n	g
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	n	m
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	s	u
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	a	g
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	e	k	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	b	c	l
8120	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	n	o	p	b	v	l
8121	e	f	s	n	f	n	a	c	b	n	...	s	o	o	p	o	o	p	b	c	l
8122	p	k	y	n	f	y	f	c	n	b	...	k	w	w	p	w	o	e	w	v	l
8123	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	o	c	l

8124 rows × 23 columns

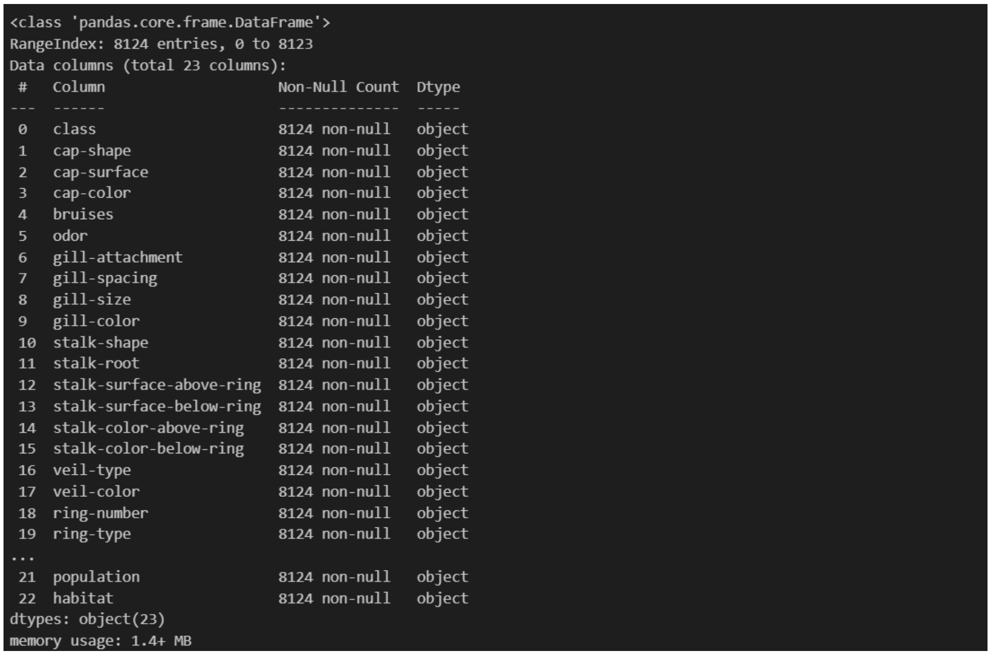
2.1.3 Getting Started Analyzing Data in Python

Initial data analysis began by identifying the total number of rows and columns in the dataset.

Figure 2.4: Displaying the total number of rows and columns in the dataset.



```
data.info()
```

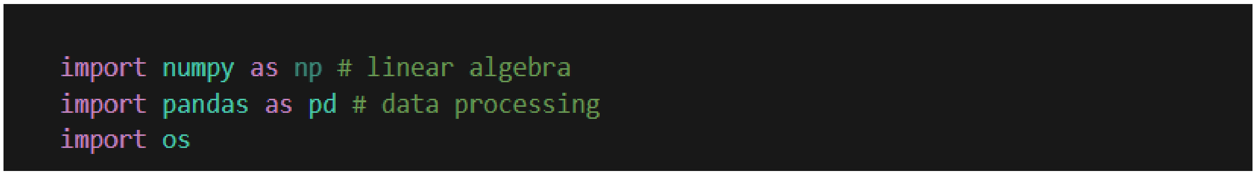


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   class                     8124 non-null   object
 1   cap-shape                 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
...
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
memory usage: 1.4+ MB
```

2.1.4 Python Packages for Data Science

We import the necessary libraries for data handling and preprocessing:

Figure 2.5: Import necessary libraries



```
import numpy as np # linear algebra
import pandas as pd # data processing
import os
```

1. **Numpy**

- Used to import the mushroom dataset from a CSV file.
- Displayed all columns to verify successful data loading.
- Checked dataset structure, data types, and missing values.
- Supported basic data manipulation and preprocessing tasks

2. **Pandas**

- used to handle numerical operations during data preprocessing.
- Supported representation of missing values using NaN.
- Assisted in preparing data for further analysis.

3. **Os**

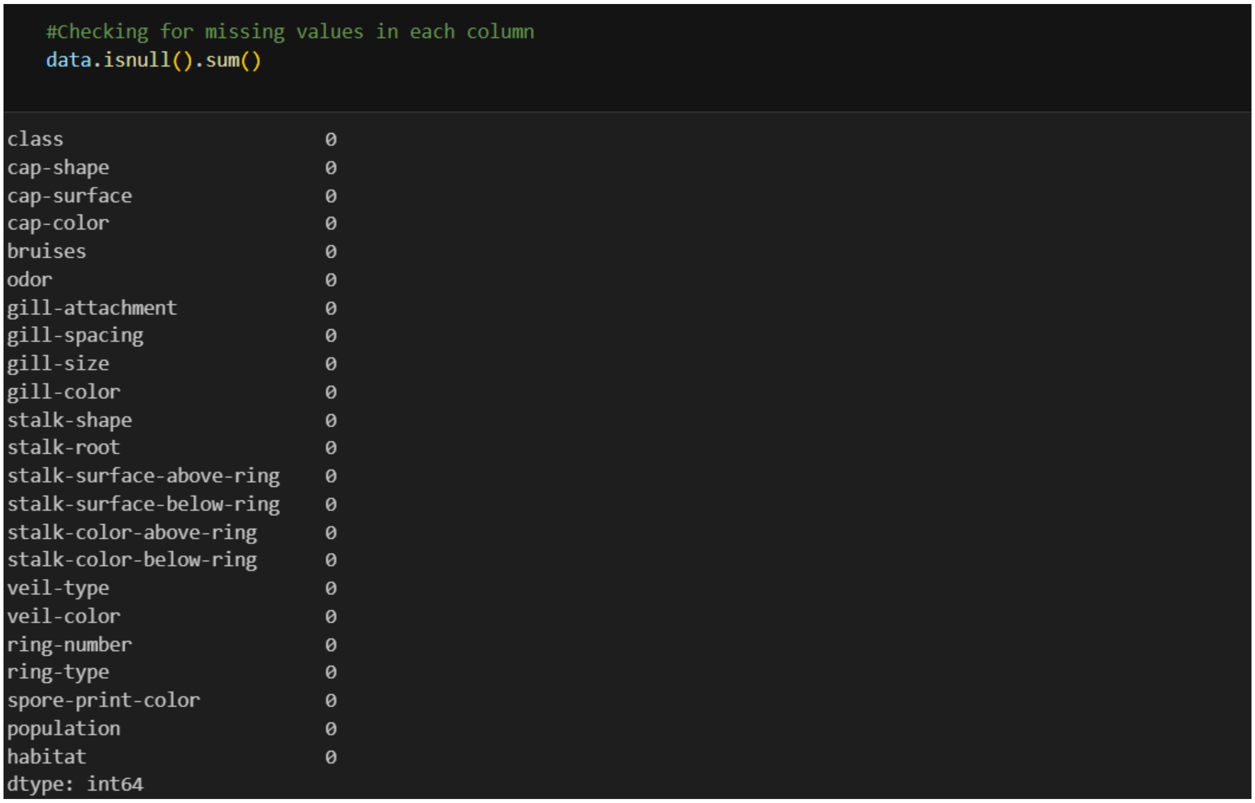
- Used to verify the presence of the dataset file in the working directory.
- Assisted in defining and managing the dataset file path before importing the data.

2.2 Data Wrangling

2.2.1 Identifying and Handling Missing Value

A thorough inspection was conducted to identify missing or null values in the dataset. After examination using **data.isnull.sum()**, it was confirmed that the mushroom dataset contains no missing values. All attributes are fully populated, allowing the dataset to be used directly without requiring data imputation or removal of records.

Figure 2.6: Display the result for confirmation of no missing values in each column.



```
#Checking for missing values in each column
data.isnull().sum()
```

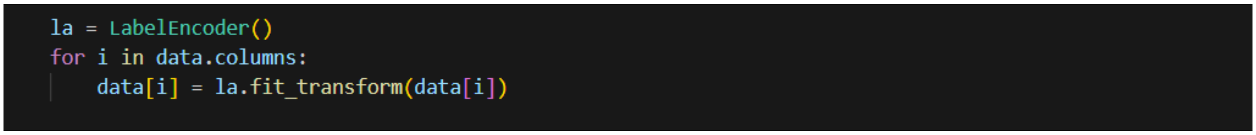
```
class                       0
cap-shape                   0
cap-surface                 0
cap-color                   0
bruises                     0
odor                        0
gill-attachment             0
gill-spacing                0
gill-size                   0
gill-color                  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
```

2.2.2 Data Formatting

The data types of each attribute were examined to ensure consistency and compatibility with machine learning algorithms. Since the mushroom dataset consists entirely of categorical variables, Label Encoding was applied to convert all categorical attributes into numerical values. This process transformed each category into integer codes, ensuring a uniform numerical representation across all features and enabling the dataset to be directly used for model training.

We use Label Encoding for simplicity:

Figure 2.7: Convert dataset string to integer



```
la = LabelEncoder()
for i in data.columns:
    data[i] = la.fit_transform(data[i])
```

Figure 2.7 shows the target column class is encoded as 0 = edible, 1 = poisonous. Each categorical feature is converted into integer codes.