FACULTY OF COMPUTING
UTM Johor Bahru

SECB3203-01(PROGRAMMING FOR BIOINFORMATIC)

Semester 01 2025/2026

Section 01

Progress 3

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

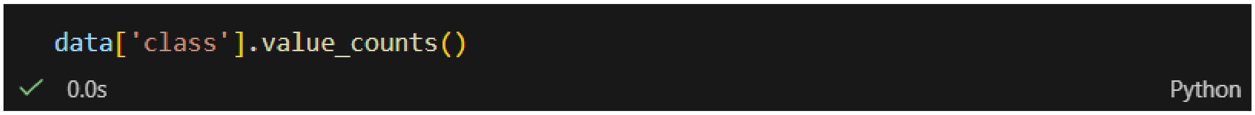
3.0 Exploratory Data Analysis

This section is focusing on understanding the data structure and the attributes' relationships. This helps in identifying important attributes of the data for the model. The results gained from this analysis will contribute to develop an effective model.

3.1 Target Variable Distribution Analysis

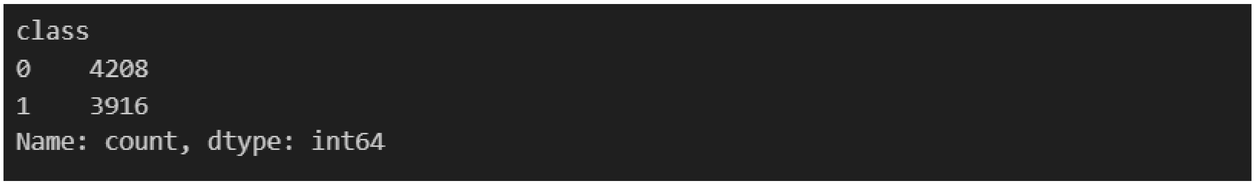
Examine the distribution of class (target variable) balance to avoid biased models. The distribution was identified using the value_counts() function, which give the total of each class (edible and poisonous)

Figure 3.0: value_counts() shows the class distribution



```python
data['class'].value_counts()
```

Result:



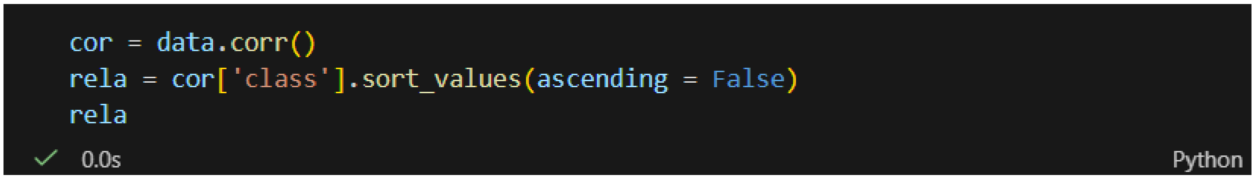
```
class
0    4208
1    3916
Name: count, dtype: int64
```

Figure 3.1: The analysis shows that the class's distribution is effectively balanced with the ratio as 1.07:1. The '0' is for edible, while '1' is for poisonous class. This is crucial for a classification because the imbalance dataset may lead to model bias, causing the algorithm to favor the majority class, affecting the result.

3.2 Correlation Analysis

To understand the linear relationships between all attributes and their individual effect on the target variable. A correlation matrix was computed. Using the data.corr() function to measure the relationship strength between two attributes. This implies all the attributes. The result is a value in between -1 and 1, where 1 indicates a perfect positive correlation, -1 for a perfect negative correlation, and 0 indicates no linear correlation. Meanwhile, sort_values(ascending = False) being used to sort the correlation values in descending order.

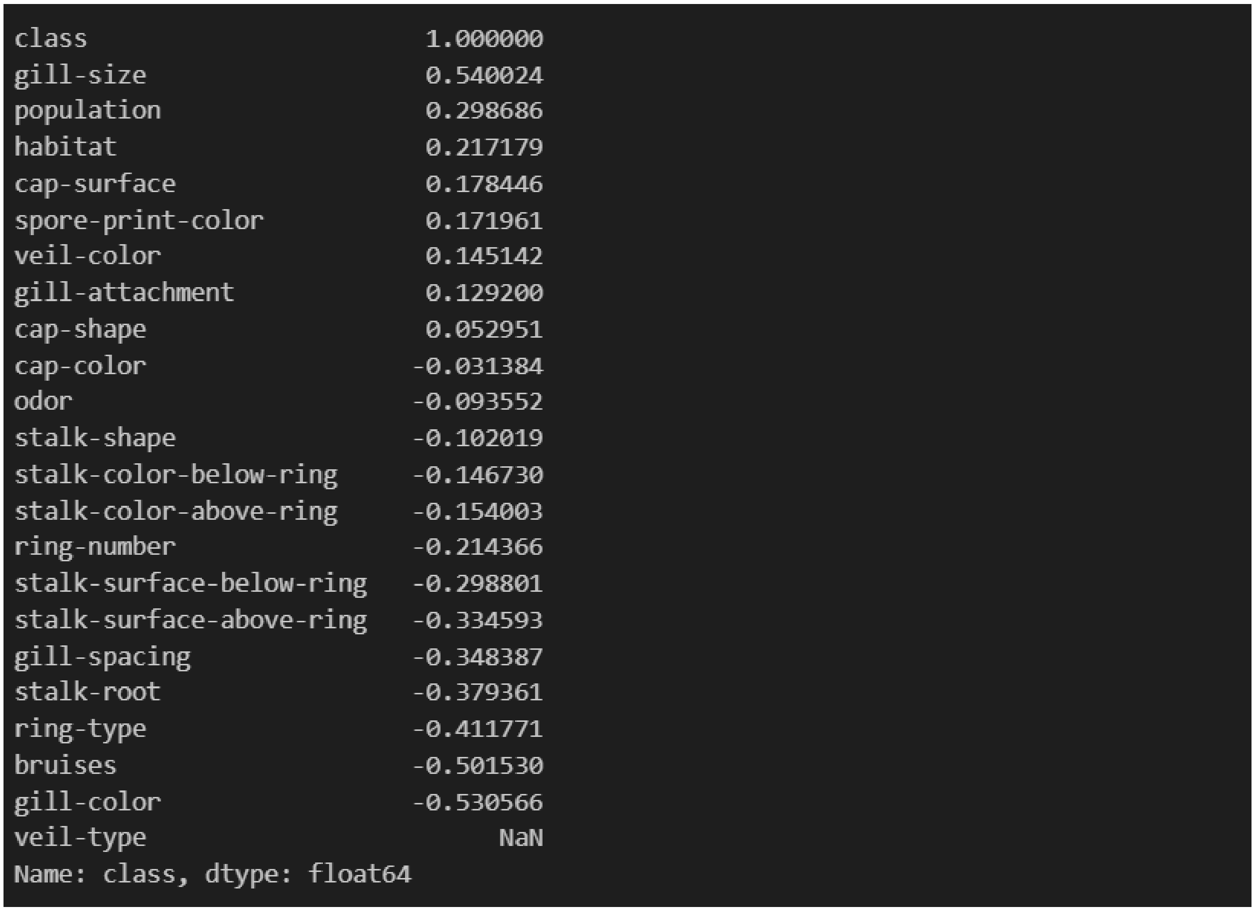
Figure 3.2: Calculate the correlation



```python
cor = data.corr()
rela = cor['class'].sort_values(ascending = False)
rela
```

Result:

Figure 3.3: The correlation output



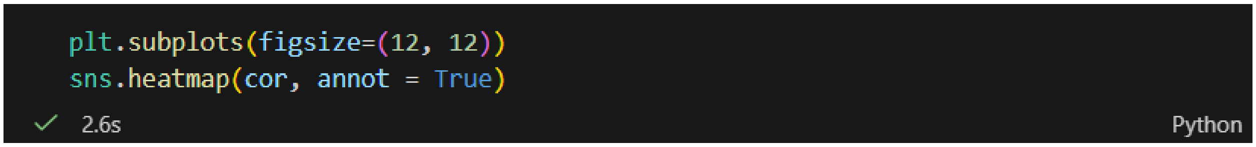
```
class                       1.000000
gill-size                   0.540024
population                  0.298686
habitat                     0.217179
cap-surface                 0.178446
spore-print-color           0.171961
veil-color                  0.145142
gill-attachment             0.129200
cap-shape                   0.052951
cap-color                  -0.031384
odor                       -0.093552
stalk-shape                -0.102019
stalk-color-below-ring     -0.146730
stalk-color-above-ring     -0.154003
ring-number                -0.214366
stalk-surface-below-ring   -0.298801
stalk-surface-above-ring   -0.334593
gill-spacing               -0.348387
stalk-root                 -0.379361
ring-type                  -0.411771
bruises                    -0.501530
gill-color                 -0.530566
veil-type                        NaN
Name: class, dtype: float64
```

Based on the output, the most predictive features are gill-size (0.54), gill-color (-0.53), and bruises (-0.50) where the gill-size is positive correlation and opposite for gill-color and bruises. It is because these features have strong correlations which means the value is greater than 0.5, crucial in predicting the mushroom class. However, the veil-type is NaN (Not a Number) because it has no variation which makes it useless for the prediction. Therefore, veil-type cannot be used to differentiate the mushroom classes.

3.3 Visualization Correlation

Using heatmap visualization to easily interpret the correlation matrix. Allowing a quick overview of all relationships between all features. The heatmap uses color gradients to differentiate the intensity of the correlation. The lightest color represents -1 while the darkest color represents +1. Thus, it is easier to identify the most influential features.

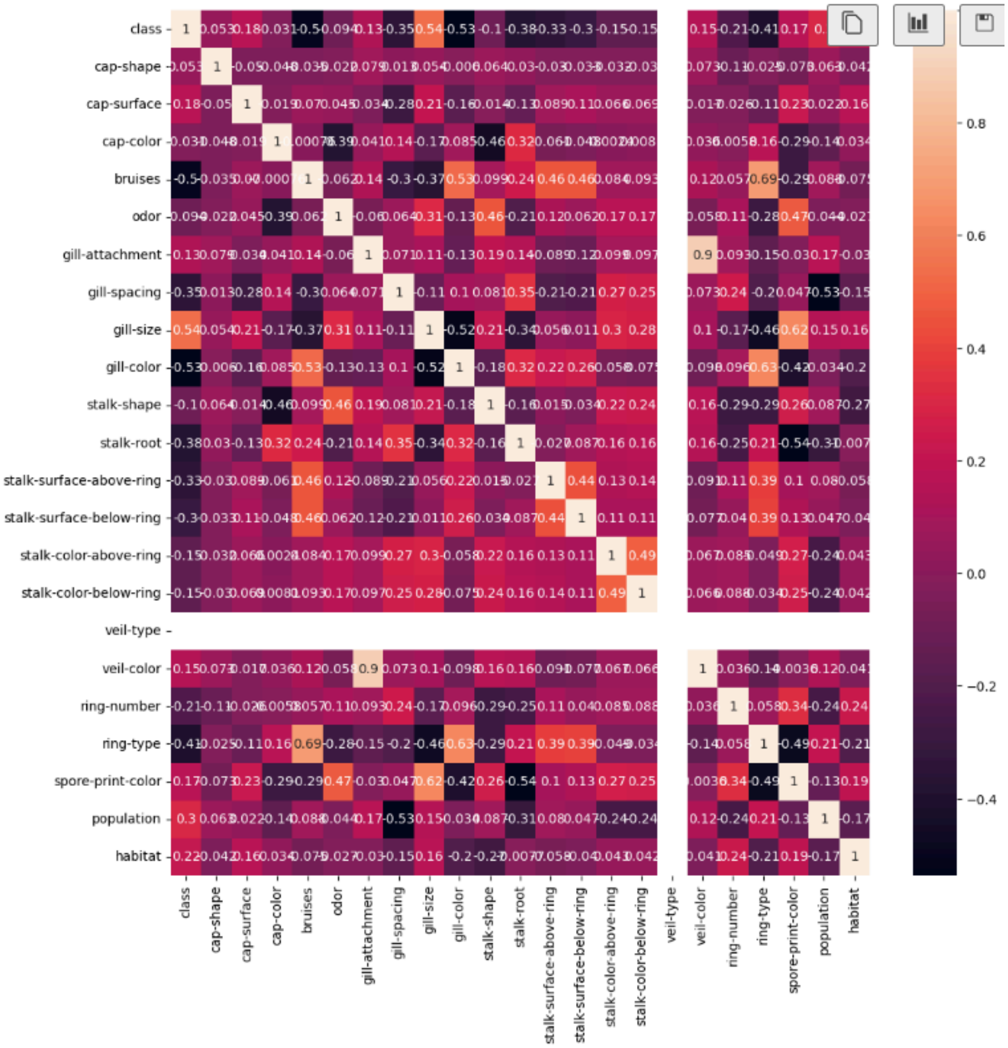
Figure 3.4: Functions to construct the heatmap visualization



```
plt.subplots(figsize=(12, 12))
sns.heatmap(cor, annot = True)
```

Result:

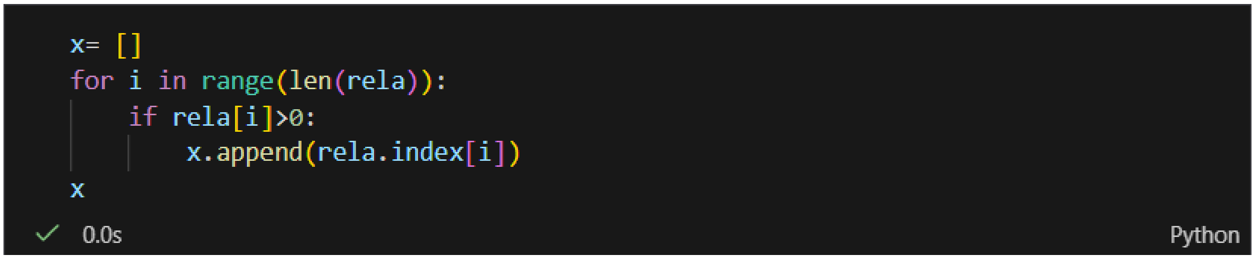
The heatmap highlighting the gill-size (0.54), gill-color (-0.53), and bruises (-0.50) among the most highly correlated with the class, a target variable. This proves that these features are strong indicators of whether a mushroom is edible or poisonous.



3.4 Feature Selection based on Correlation

Following the correlation analysis, a features selection process was conducted to extract the positive correlated variables and prepare the final dataset for model training. The objective is to create a dataset that would enhance model interpretability and efficiency, helping the model to find a perfect algorithm and gain 100% accuracy.

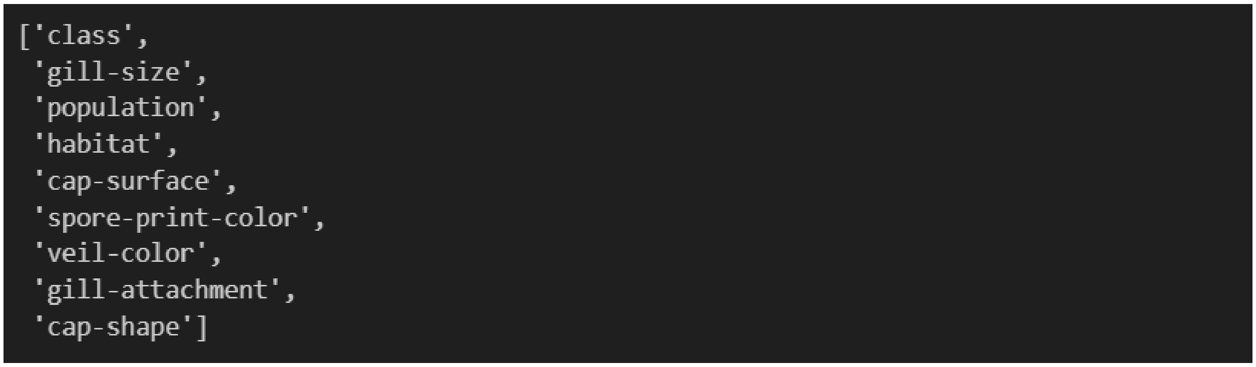
Figure 3.5: Selected variables with positive correlation



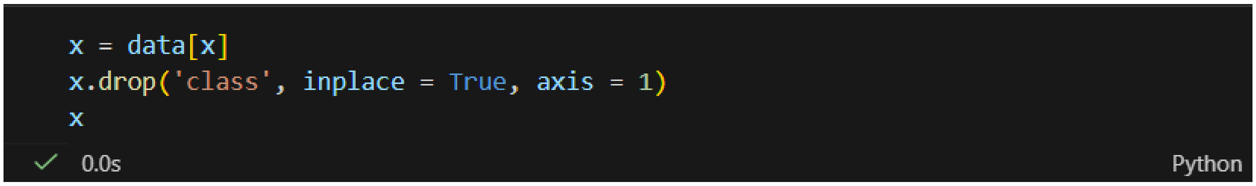
```python
x= []
for i in range(len(rela)):
    if rela[i]>0:
        x.append(rela.index[i])
x
```

Result:

Figure 3.6: The selected variables for model training and testing

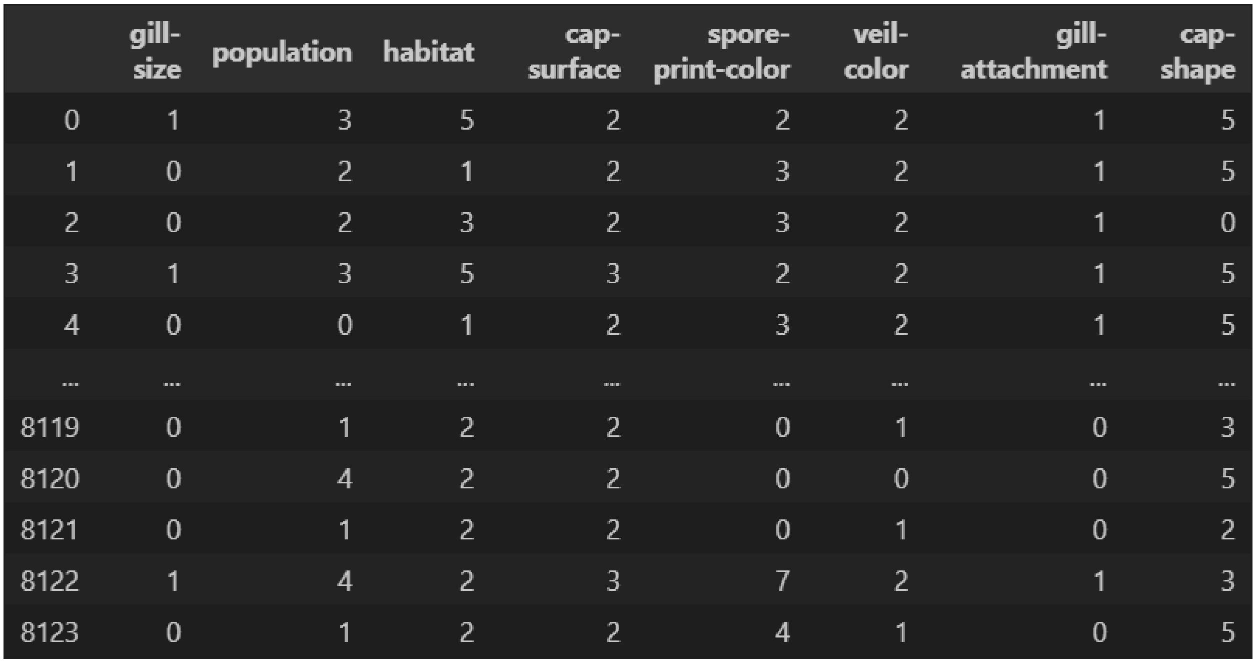


```
['class',
 'gill-size',
 'population',
 'habitat',
 'cap-surface',
 'spore-print-color',
 'veil-color',
 'gill-attachment',
 'cap-shape']
```



```
x = data[x]
x.drop('class', inplace = True, axis = 1)
x
```

Result:

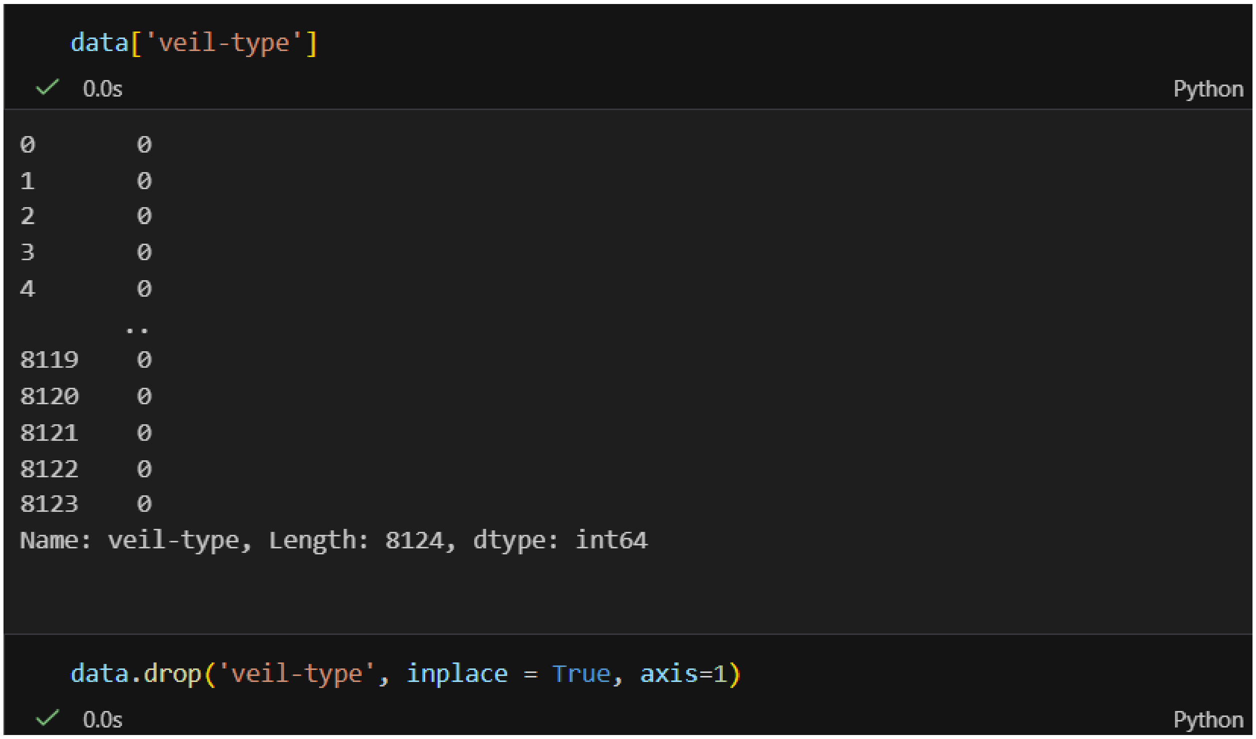


	gill-size	population	habitat	cap-surface	spore-print-color	veil-color	gill-attachment	cap-shape
0	1	3	5	2	2	2	1	5
1	0	2	1	2	3	2	1	5
2	0	2	3	2	3	2	1	0
3	1	3	5	3	2	2	1	5
4	0	0	1	2	3	2	1	5
...	...	...	...	...	...	...	...	...
8119	0	1	2	2	0	1	0	3
8120	0	4	2	2	0	0	0	5
8121	0	1	2	2	0	1	0	2
8122	1	4	2	3	7	2	1	3
8123	0	1	2	2	4	1	0	5

Result Interpretation:

Above are the variables (columns) that have positive correlation value extracted for model training. By selecting these variables, the model is built exclusively on characteristics that are expressing the poisonous class. Therefore, the model is trained to be highly aware to poisonous indicators and prioritize any feature that strongly associated with toxicity. This minimizes the False Negative where the model classifies a poisonous mushroom as edible.

Figure 3.7: Shows the veil-type attributes have the same value for all instances (‘0’ refers to ‘partial’)



```
data['veil-type']
```

```
0       0
1       0
2       0
3       0
4       0
       ..
8119    0
8120    0
8121    0
8122    0
8123    0
Name: veil-type, Length: 8124, dtype: int64
```

```
data.drop('veil-type', inplace = True, axis=1)
```

Therefore, the veil-type was removed from the dataset because it was completely useless in the model training. It doesn’t help the model to differentiate between edible or poisonous mushrooms.