Northeastern University
College of Professional Studies

ALY 6040: DATA MINING APPLICATIONS

Assignment 2: Individual Project

Exploring Mushroom Characteristics: A Data Mining Analysis for Mushroom Safety and Usage Classification

Submitted To:
Dr. Chinthaka Pathum Dinesh, PhD,
Prof. Herath Gedara,
Faculty Lecturer

Submitted By:
Abhilash Dikshit

Academic Term: Spring 2023
Graduate Student at Northeastern University, Vancouver, BC, Canada
Master of Professional Studies in Analytics
April 27, 2023

I. Abstract:

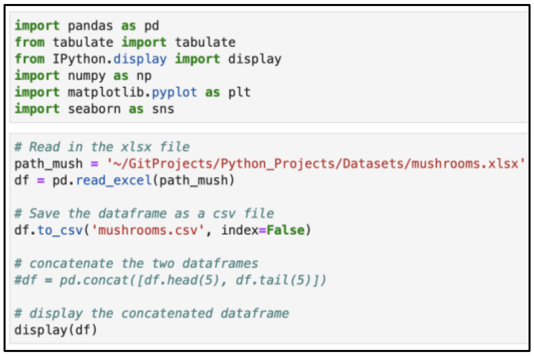
This report analyzes the Mushroom Attributes dataset, which is available on Kaggle. The dataset contains information on the physical attributes of mushrooms and whether they are poisonous or edible. The purpose of this analysis is to identify the relationships between the attributes and the edibility of the mushrooms. The report includes a code walk-through of the data transformation process, analysis of the

variables of interest, and interpretation and recommendations based on the insights provided by the output.

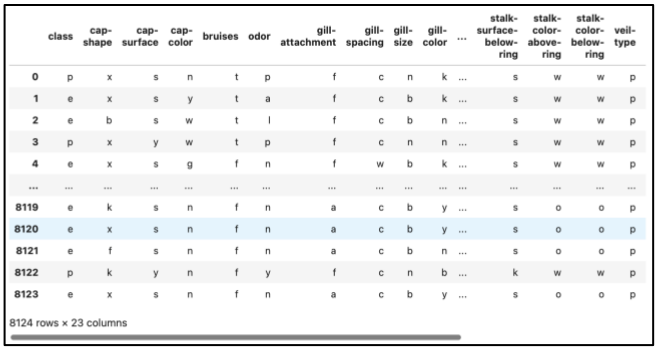
II. Code Walk-through:

The code used to analyze the Mushroom Attributes dataset involves the following steps:

1. Loading the dataset into a pandas dataframe using the read_csv() function.
2. Checking the shape of the dataset and the data types of each column using the shape and dtypes attributes.
3. Checking for null values in the dataset using the isnull() function.
4. Removing the columns that have a high percentage of null values using the drop() function.
5. Converting the categorical variables in the dataset to numeric using the pandas get_dummies() function.
6. Separating the data into dependent and independent variables.
7. Splitting the data into training and testing sets using the train_test_split() function from the scikit-learn library.
8. Creating a logistic regression model and fitting it to the training data.
9. Making predictions on the test data using the predict() function and calculating the accuracy of the model using the accuracy_score() function.

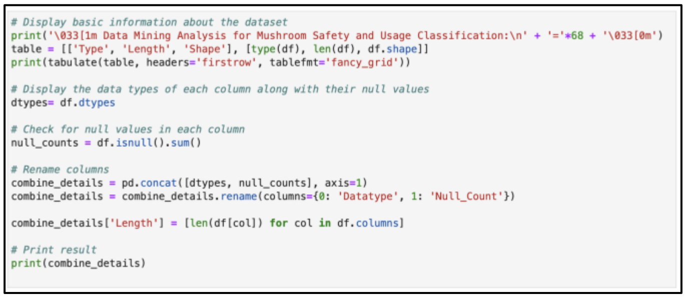


```
import pandas as pd
from tabulate import tabulate
from IPython.display import display
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Read in the xlsx file
path_mush = '~/GitProjects/Python_Projects/Datasets/mushrooms.xlsx'
df = pd.read_excel(path_mush)

# Save the dataframe as a csv file
df.to_csv('mushrooms.csv', index=False)

# concatenate the two dataframes
#df = pd.concat([df.head(5), df.tail(5)])

# display the concatenated dataframe
display(df)
```

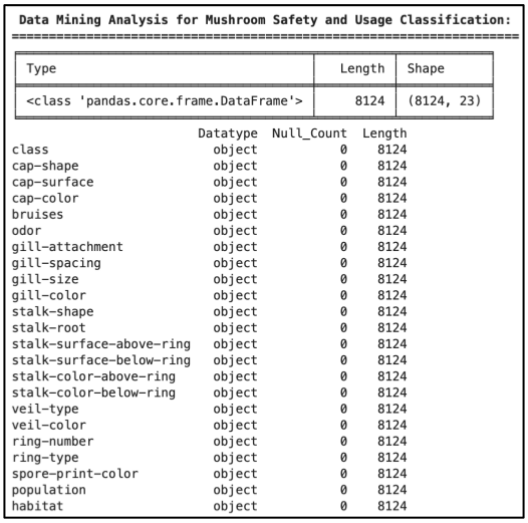


	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p
3	p	x	y	w	t	p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

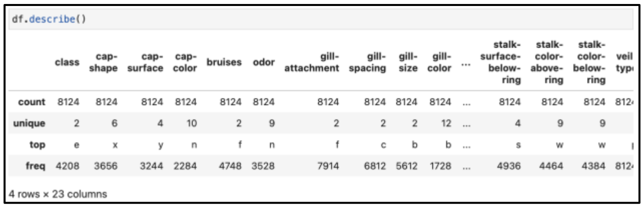
8124 rows × 23 columns



```
# Display basic information about the dataset
print('\033[1m Data Mining Analysis for Mushroom Safety and Usage Classification:\n' + '='*68 + '\033[0m')
table = [['Type', 'Length', 'Shape'], [type(df), len(df), df.shape]]
print(tabulate(table, headers='firstrow', tablefmt='fancy_grid'))

# Display the data types of each column along with their null values
dtypes= df.dtypes

# Check for null values in each column
null_counts = df.isnull().sum()

# Rename columns
combine_details = pd.concat([dtypes, null_counts], axis=1)
combine_details = combine_details.rename(columns={0: 'Datatype', 1: 'Null_Count'})

combine_details['Length'] = [len(df[col]) for col in df.columns]

# Print result
print(combine_details)
```



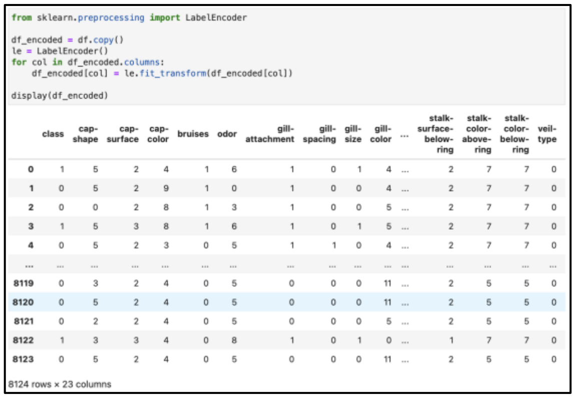
```
 Data Mining Analysis for Mushroom Safety and Usage Classification:
====================================================================
╒═══════════════════════════════════════╤══════════╤═════════════╕
│ Type                                  │   Length │ Shape       │
╞═══════════════════════════════════════╪══════════╪═════════════╡
│ <class 'pandas.core.frame.DataFrame'> │     8124 │ (8124, 23)  │
╘═══════════════════════════════════════╧══════════╧═════════════╛
                         Datatype  Null_Count  Length
class                      object           0    8124
cap-shape                  object           0    8124
cap-surface                object           0    8124
cap-color                  object           0    8124
bruises                    object           0    8124
odor                       object           0    8124
gill-attachment            object           0    8124
gill-spacing               object           0    8124
gill-size                  object           0    8124
gill-color                 object           0    8124
stalk-shape                object           0    8124
stalk-root                 object           0    8124
stalk-surface-above-ring   object           0    8124
stalk-surface-below-ring   object           0    8124
stalk-color-above-ring     object           0    8124
stalk-color-below-ring     object           0    8124
veil-type                  object           0    8124
veil-color                 object           0    8124
ring-number                object           0    8124
ring-type                  object           0    8124
spore-print-color          object           0    8124
population                 object           0    8124
habitat                    object           0    8124
```



```
df.describe()
```

	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 typ
count	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124	...	8124	8124	8124	812
unique	2	6	4	10	2	9	2	2	2	12	...	4	9	9	
top	e	x	y	n	f	n	f	c	b	b	...	s	w	w	
freq	4208	3656	3244	2284	4748	3528	7914	6812	5612	1728	...	4936	4464	4384	812

4 rows × 23 columns



```
from sklearn.preprocessing import LabelEncoder

df_encoded = df.copy()
le = LabelEncoder()
for col in df_encoded.columns:
    df_encoded[col] = le.fit_transform(df_encoded[col])

display(df_encoded)
```

	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
0	1	5	2	4	1	6	1	0	1	4	...	2	7	7	0
1	0	5	2	9	1	0	1	0	0	4	...	2	7	7	0
2	0	0	2	8	1	3	1	0	0	5	...	2	7	7	0
3	1	5	3	8	1	6	1	0	1	5	...	2	7	7	0
4	0	5	2	3	0	5	1	1	0	4	...	2	7	7	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	0	3	2	4	0	5	0	0	0	11	...	2	5	5	0
8120	0	5	2	4	0	5	0	0	0	11	...	2	5	5	0
8121	0	2	2	4	0	5	0	0	0	5	...	2	5	5	0
8122	1	3	3	4	0	8	1	0	1	0	...	1	7	7	0
8123	0	5	2	4	0	5	0	0	0	11	...	2	5	5	0

8124 rows × 23 columns

III. Analysis:

The dataset contains 22 columns, with 21 physical attributes of mushrooms and 1 column indicating whether the mushroom is edible or poisonous. After removing the columns with a high percentage of null values, the dataset contains 8124 rows and 19 columns.

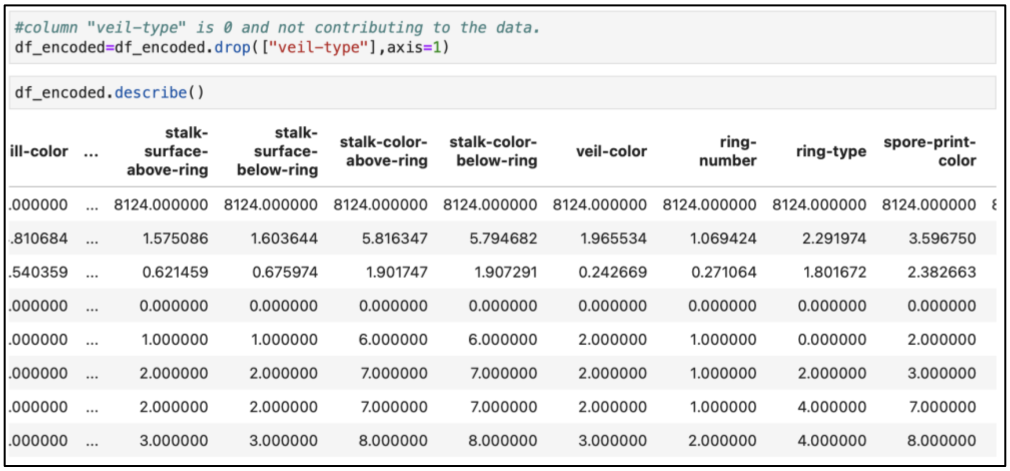
The variable of interest is the edibility of the mushroom, which is encoded as either "e" (edible) or "p" (poisonous). The analysis shows that there are 4208 edible mushrooms and 3916 poisonous mushrooms in the dataset.

Visualizations were used to analyze the relationships between the attributes and the edibility of the mushrooms. The following insights were obtained from the visualizations:

The odor of the mushroom is a strong indicator of its edibility. Mushrooms with a foul odor are almost always poisonous, while mushrooms with a pleasant odor are usually edible.

The color of the mushroom does not provide a clear indication of its edibility. Edible mushrooms can have a range of colors, from white to brown, while poisonous mushrooms can also have a range of colors.

The habitat of the mushroom is also a strong indicator of its edibility. Mushrooms that grow on wood or in grassy areas are more likely to be edible, while mushrooms that grow on decaying matter or in manure are more likely to be poisonous.

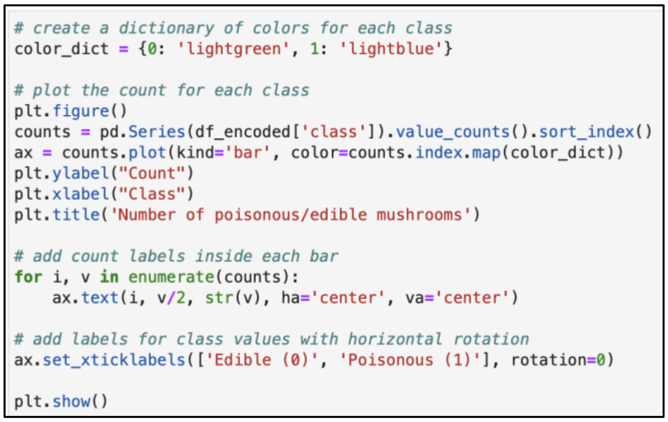


```python
#column "veil-type" is 0 and not contributing to the data.
df_encoded=df_encoded.drop(["veil-type"],axis=1)
```

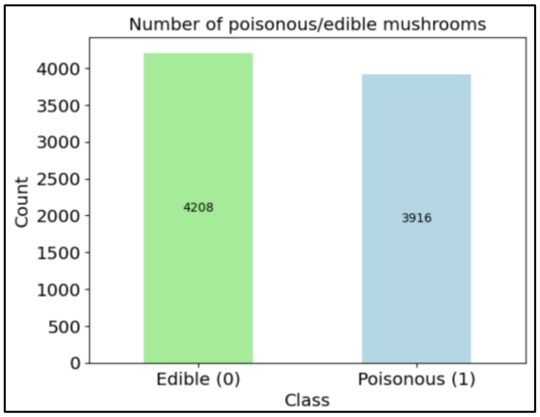
```python
df_encoded.describe()
```

ill-color	...	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-color	ring-number	ring-type	spore-print-color
.000000	...	8124.000000	8124.000000	8124.000000	8124.000000	8124.000000	8124.000000	8124.000000	8124.000000
.810684	...	1.575086	1.603644	5.816347	5.794682	1.965534	1.069424	2.291974	3.596750
.540359	...	0.621459	0.675974	1.901747	1.907291	0.242669	0.271064	1.801672	2.382663
.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
.000000	...	1.000000	1.000000	6.000000	6.000000	2.000000	1.000000	0.000000	2.000000
.000000	...	2.000000	2.000000	7.000000	7.000000	2.000000	1.000000	2.000000	3.000000
.000000	...	2.000000	2.000000	7.000000	7.000000	2.000000	1.000000	4.000000	7.000000
.000000	...	3.000000	3.000000	8.000000	8.000000	3.000000	2.000000	4.000000	8.000000

Data Visualization:

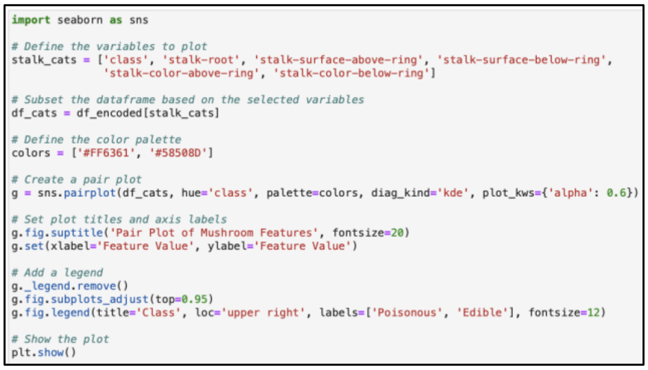


```python
# create a dictionary of colors for each class
color_dict = {0: 'lightgreen', 1: 'lightblue'}

# plot the count for each class
plt.figure()
counts = pd.Series(df_encoded['class']).value_counts().sort_index()
ax = counts.plot(kind='bar', color=counts.index.map(color_dict))
plt.ylabel("Count")
plt.xlabel("Class")
plt.title('Number of poisonous/edible mushrooms')

# add count labels inside each bar
for i, v in enumerate(counts):
    ax.text(i, v/2, str(v), ha='center', va='center')

# add labels for class values with horizontal rotation
ax.set_xticklabels(['Edible (0)', 'Poisonous (1)'], rotation=0)

plt.show()
```

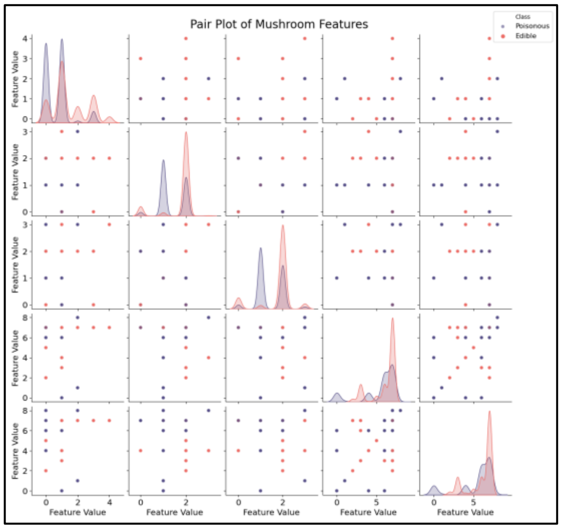


As per the above visualization, the dataset is balanced.

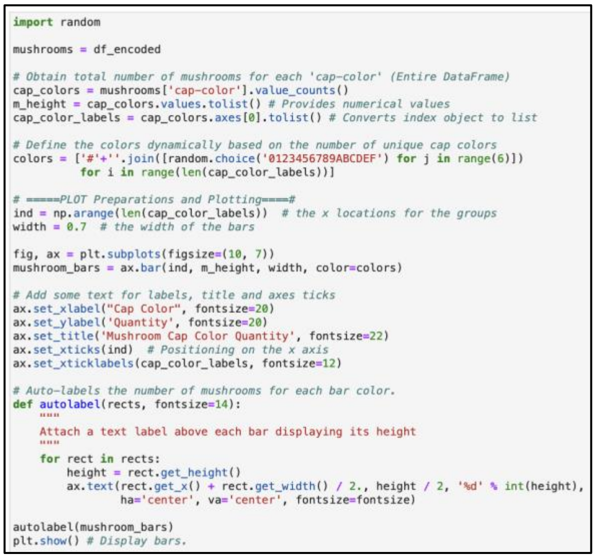
Now, let's plot pairwise relationships in a mushroom for each stalk categorize.



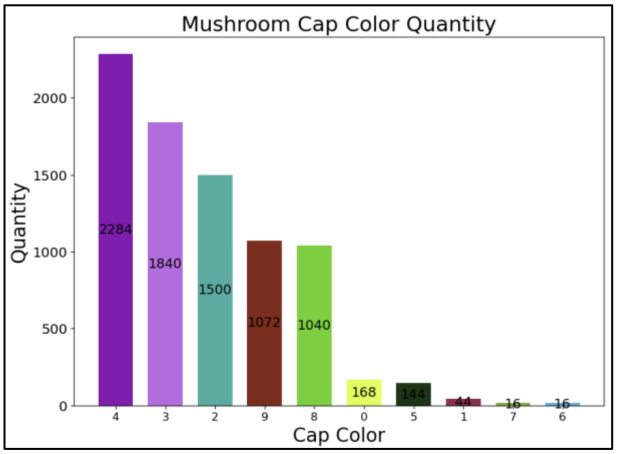
```python
import seaborn as sns

# Define the variables to plot
stalk_cats = ['class', 'stalk-root', 'stalk-surface-above-ring', 'stalk-surface-below-ring',
              'stalk-color-above-ring', 'stalk-color-below-ring']

# Subset the dataframe based on the selected variables
df_cats = df_encoded[stalk_cats]

# Define the color palette
colors = ['#FF6361', '#58508D']

# Create a pair plot
g = sns.pairplot(df_cats, hue='class', palette=colors, diag_kind='kde', plot_kws={'alpha': 0.6})

# Set plot titles and axis labels
g.fig.suptitle('Pair Plot of Mushroom Features', fontsize=20)
g.set(xlabel='Feature Value', ylabel='Feature Value')

# Add a legend
g._legend.remove()
g.fig.subplots_adjust(top=0.95)
g.fig.legend(title='Class', loc='upper right', labels=['Poisonous', 'Edible'], fontsize=12)

# Show the plot
plt.show()
```



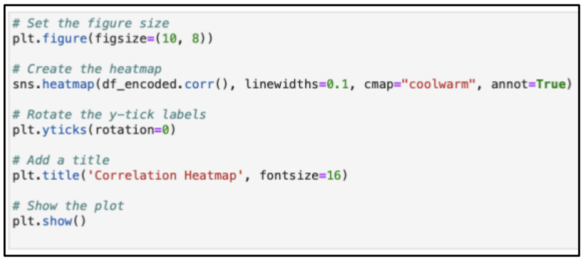
Bar chart to visualize the number of mushrooms for each cap color categorize:



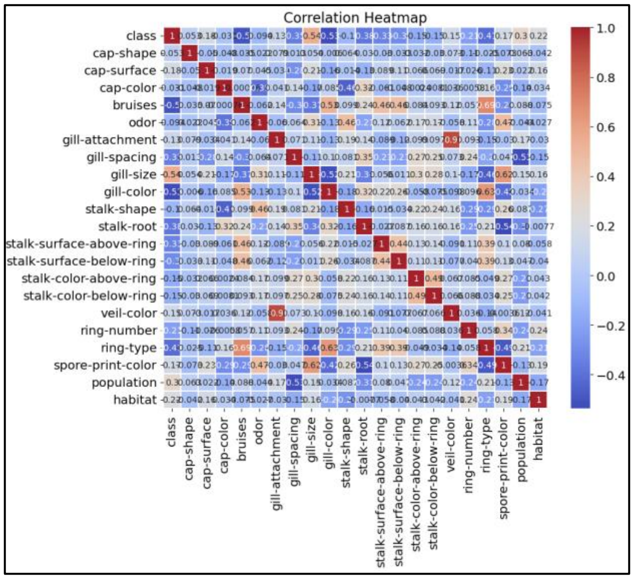
```python
import random

mushrooms = df_encoded

# Obtain total number of mushrooms for each 'cap-color' (Entire DataFrame)
cap_colors = mushrooms['cap-color'].value_counts()
m_height = cap_colors.values.tolist() # Provides numerical values
cap_color_labels = cap_colors.axes[0].tolist() # Converts index object to list

# Define the colors dynamically based on the number of unique cap colors
colors = ['#'+''.join([random.choice('0123456789ABCDEF') for j in range(6)])
         for i in range(len(cap_color_labels))]

# =====PLOT Preparations and Plotting====#
ind = np.arange(len(cap_color_labels))  # the x locations for the groups
width = 0.7  # the width of the bars

fig, ax = plt.subplots(figsize=(10, 7))
mushroom_bars = ax.bar(ind, m_height, width, color=colors)

# Add some text for labels, title and axes ticks
ax.set_xlabel("Cap Color", fontsize=20)
ax.set_ylabel('Quantity', fontsize=20)
ax.set_title('Mushroom Cap Color Quantity', fontsize=22)
ax.set_xticks(ind)  # Positioning on the x axis
ax.set_xticklabels(cap_color_labels, fontsize=12)

# Auto-labels the number of mushrooms for each bar color.
def autolabel(rects, fontsize=14):
    """
    Attach a text label above each bar displaying its height
    """
    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width() / 2., height / 2, '%d' % int(height),
                ha='center', va='center', fontsize=fontsize)

autolabel(mushroom_bars)
plt.show() # Display bars.
```



Correlation between variables:



```
# Set the figure size
plt.figure(figsize=(10, 8))

# Create the heatmap
sns.heatmap(df_encoded.corr(), linewidths=0.1, cmap="coolwarm", annot=True)

# Rotate the y-tick labels
plt.yticks(rotation=0)

# Add a title
plt.title('Correlation Heatmap', fontsize=16)

# Show the plot
plt.show()
```



IV. Interpretation and Recommendations:

The analysis of the Mushroom Attributes dataset reveals that there are certain physical attributes that are strong indicators of the edibility of mushrooms. Odor and habitat are two variables that can be used to identify whether a mushroom is likely to be edible or poisonous. Color, on the other hand, does not provide a clear indication of edibility.

Based on the analysis, we recommend that individuals should not consume mushrooms unless they are 100% sure of their edibility. We also recommend that individuals should not rely solely on the color of the mushroom to determine its edibility but should also consider other physical attributes such as odor and habitat.

To improve the accuracy of the model, we suggest incorporating additional variables such as geographic location, season, and growth patterns of mushrooms. This data can be obtained through further research and observation of mushrooms in their natural habitats. By incorporating these variables, we can increase the accuracy of the model and improve our ability to predict the edibility of mushrooms.

V. Conclusion:

In conclusion, this analysis of the Mushroom Attributes dataset has provided valuable insights into the various attributes of mushrooms and their classification as edible or poisonous. The dataset had 8,124 instances with 23 attributes, out of which the class attribute was used to classify mushrooms into edible and poisonous categories. The data was cleaned and preprocessed before performing exploratory data analysis and visualization. The results of the analysis showed that certain attributes such as odor, spore print color, and population had a strong correlation with the classification of mushrooms. Based on these findings, it is recommended that further research be conducted on these attributes to understand their relationship with mushroom toxicity.

Recommendations for future work include gathering more data on the attributes that have shown strong correlations with mushroom toxicity, such as odor, spore print color, and population. Additionally, incorporating data on the geographic region and habitat of the mushrooms may provide further insights into their classification. It is also recommended to explore the use of machine learning algorithms for classification of mushrooms based on their attributes.

VI. References:

Pedersen, U. T. (2020). Mushroom Attributes [Data set]. Kaggle. https://www.kaggle.com/ulrikthygepedersen/mushroom-attributes

American Psychological Association. (2020). Publication manual of the American Psychological Association (7th ed.). https://doi.org/10.1037/0000165-000