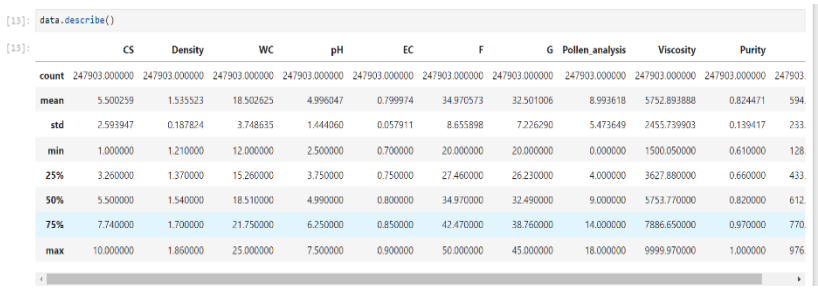
Data Collection and Preprocessing Phase

Date	4 July2024
Team ID	739767
Project Title	Honey price prediction based on purity.
Maximum Marks	6 Marks

Preprocessing Template

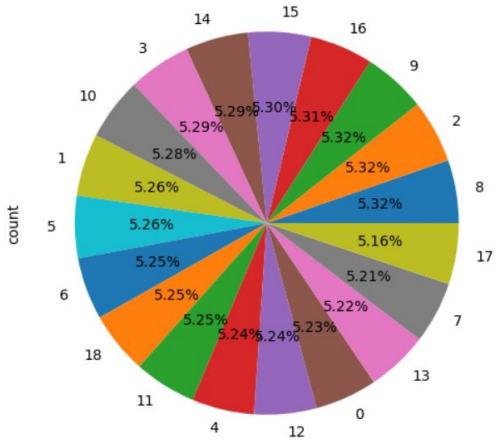
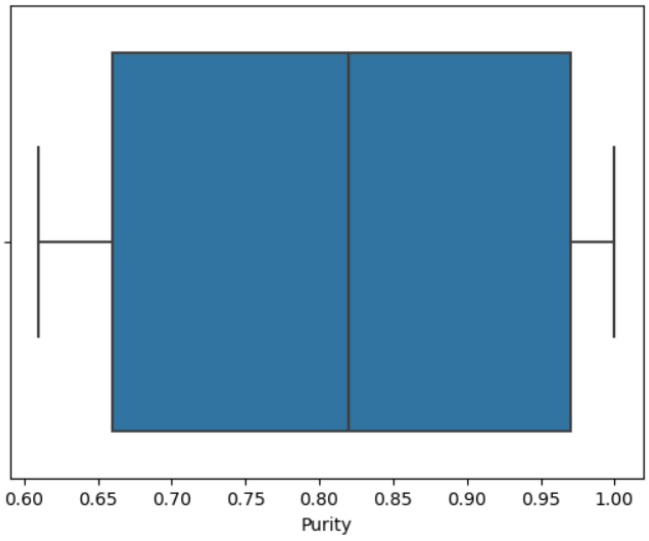
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	

```
[13]: data.describe()
```

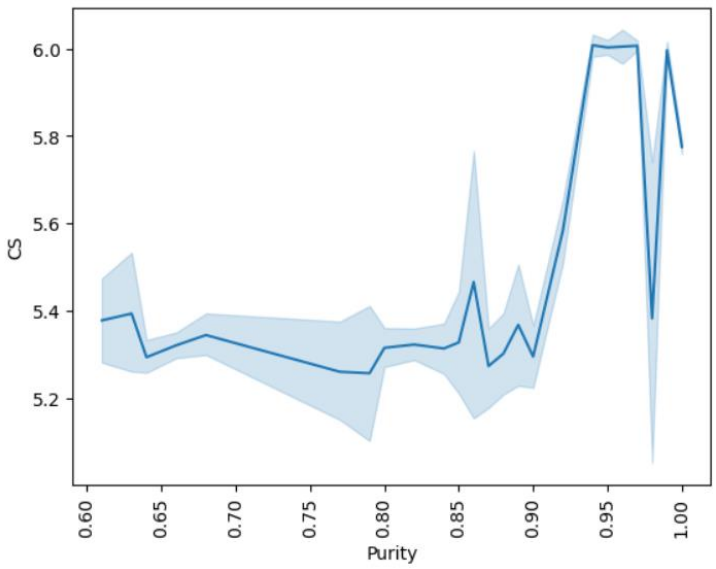
[13]:

	CS	Density	WC	pH	EC	F	G	Pollen_analysis	Viscosity	Purity	
count	247903.000000	247903.000000	247903.000000	247903.000000	247903.000000	247903.000000	247903.000000	247903.000000	247903.000000	247903.000000	247903.
mean	5.500259	1.535523	18.502625	4.996047	0.799974	34.970573	32.501006	8.993618	5752.893888	0.824471	594.
std	2.593947	0.187824	3.748635	1.444060	0.057911	8.655898	7.226290	5.473649	2455.739903	0.139417	233.
min	1.000000	1.210000	12.000000	2.500000	0.700000	20.000000	20.000000	0.000000	1500.050000	0.610000	128.
25%	3.260000	1.370000	15.260000	3.750000	0.750000	27.460000	26.230000	4.000000	3627.880000	0.660000	433.
50%	5.500000	1.540000	18.510000	4.990000	0.800000	34.970000	32.490000	9.000000	5753.770000	0.820000	612.
75%	7.740000	1.700000	21.750000	6.250000	0.850000	42.470000	38.760000	14.000000	7886.650000	0.970000	770.
max	10.000000	1.860000	25.000000	7.500000	0.900000	50.000000	45.000000	18.000000	9999.970000	1.000000	976.

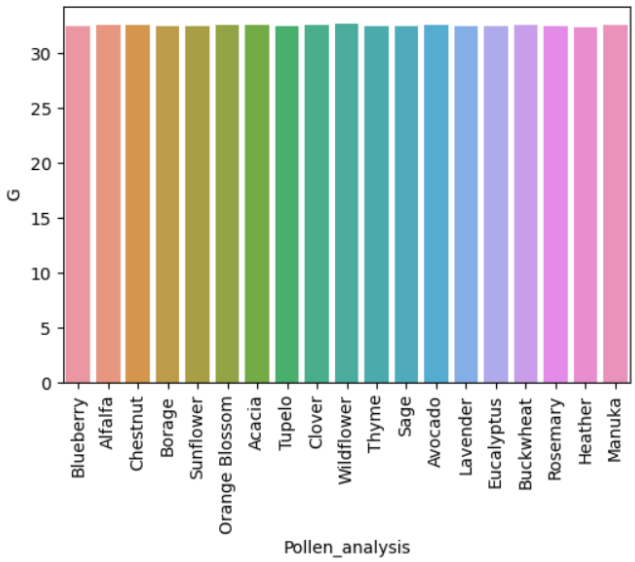
Univariate Analysis	[15]: ``` plt.figure(figsize=(6,6)) data["Pollen_analysis"].value_counts().plot(kind="pie",subplots=True,autopct="%1.2f%%") ``` [15]: ``` array([<Axes: ylabel='count'>], dtype=object) ```  [16]: ``` sns.boxplot(x="Purity",data=data) plt.show() ``` 

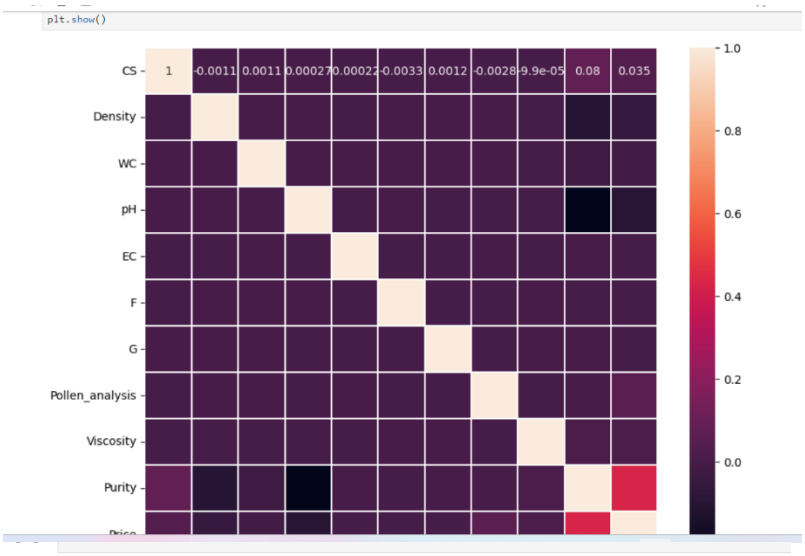
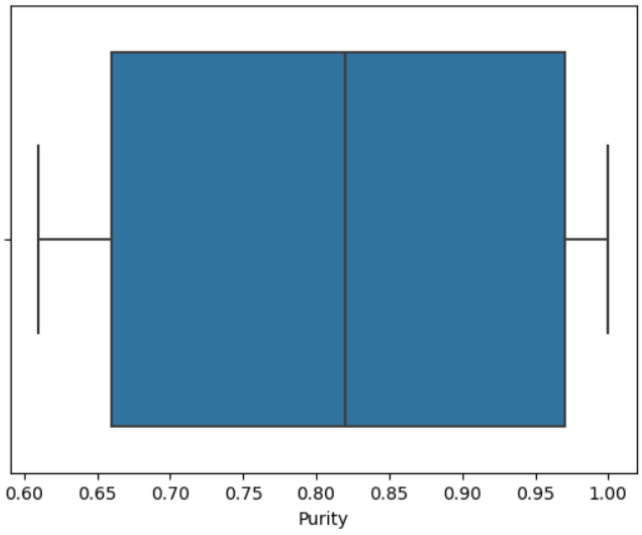
Bivariate Analysis

```
sns.lineplot(y="CS",x="Purity",data=data)
plt.xticks(rotation=90)
plt.show()
```



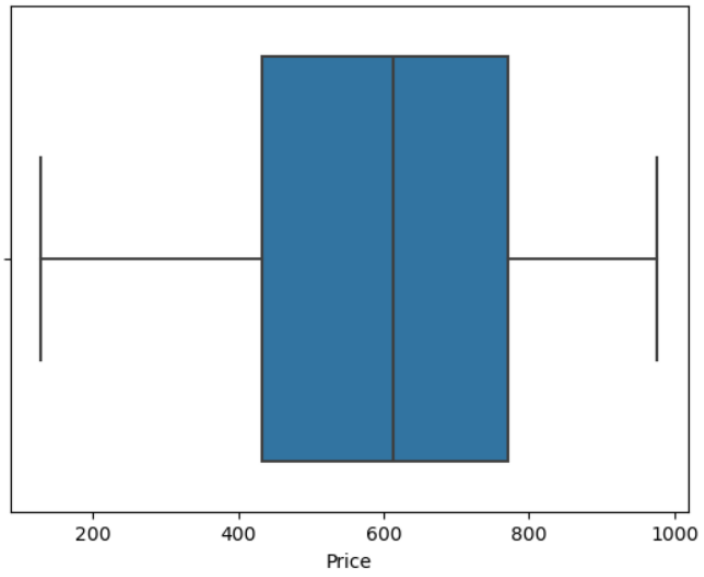

```
[7]: plt.figure(figsize=(6,4))
sns.barplot(y="G",x="Pollen_analysis",data=data,ci=None)
plt.xticks(rotation=90)
plt.show()
```



Multivariate Analysis	 [19]: sns.boxplot(x="Purity",data=data) plt.show() 

<table>
<tr><td></td><td>

```
20]: sns.boxplot(x="Price",data=data)
     plt.show()
```



</td></tr>
<tr><td>Outliers and Anomalies</td><td>-----</td></tr>
<tr><td colspan="2">Data Preprocessing Code Screenshots</td></tr>
<tr><td>Loading Data</td><td>

```
2]: data=pd.read_csv('honey_purity_dataset.csv')
    data
```

2]:

	CS	Density	WC	pH	EC	F	G	Pollen_analysis	Viscosity	Purity	Price
0	2.81	1.75	23.04	6.29	0.76	39.02	33.63	Blueberry	4844.50	0.68	645.24
1	9.47	1.82	17.50	7.20	0.71	38.15	34.41	Alfalfa	6689.02	0.89	385.85
2	4.61	1.84	23.72	7.31	0.80	27.47	34.36	Chestnut	6883.60	0.66	639.64
3	1.77	1.40	16.61	4.01	0.78	31.52	28.15	Blueberry	7167.56	1.00	946.46
4	6.11	1.25	19.63	4.82	0.90	29.65	42.52	Alfalfa	5125.44	1.00	432.62
...	...	...	...	...	...	...	...	...	...	...	...
247898	1.98	1.29	17.90	4.82	0.89	36.10	34.69	Rosemary	8261.63	1.00	754.98
247899	6.18	1.67	19.54	4.91	0.85	31.15	20.82	Acacia	6939.39	1.00	543.41
247900	7.78	1.49	15.78	5.69	0.73	44.60	44.07	Chestnut	4139.79	0.64	615.46
247901	5.78	1.74	14.96	6.81	0.83	47.19	37.79	Avocado	4417.74	0.97	949.32
247902	8.96	1.86	18.62	6.89	0.86	25.94	42.88	Lavender	8119.62	0.64	384.48

247903 rows × 11 columns

</td></tr>
</table>

Handling Null Values	12]: `data.isnull().sum()` 12]: CS 0 Density 0 WC 0 pH 0 EC 0 F 0 G 0 Pollen_analysis 0 Viscosity 0 Purity 0 Price 0 dtype: int64
Outliers	-------