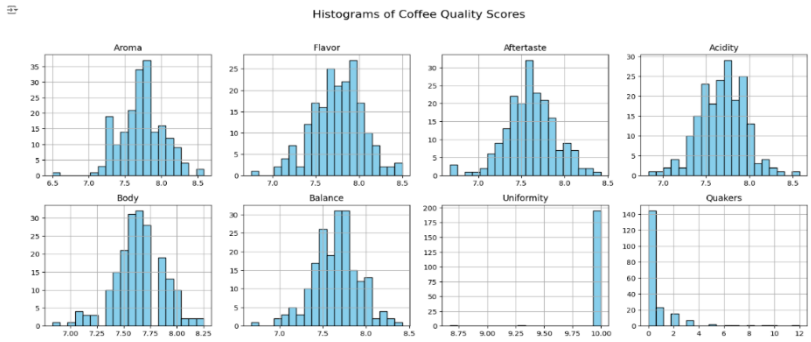
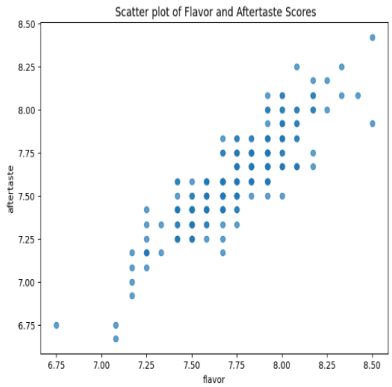
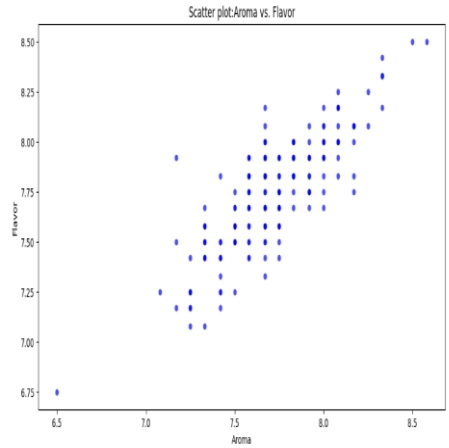
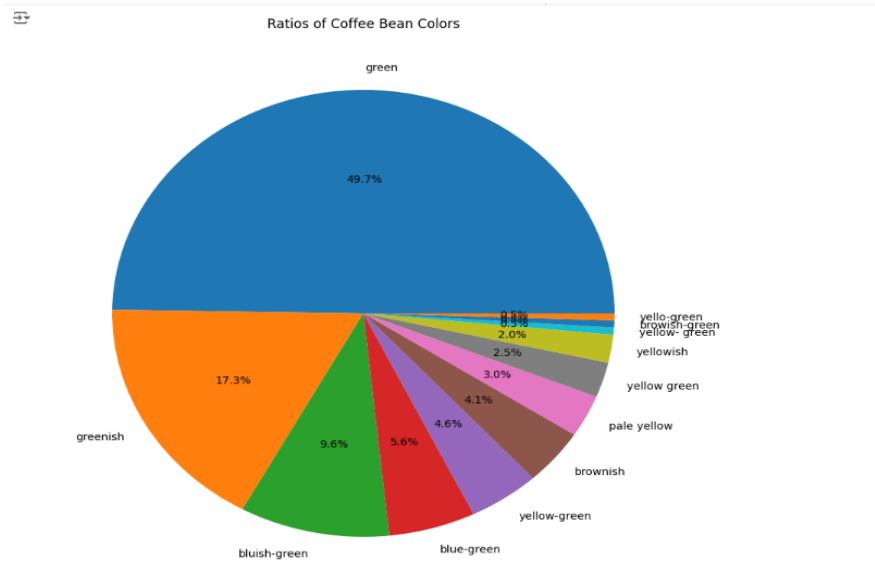
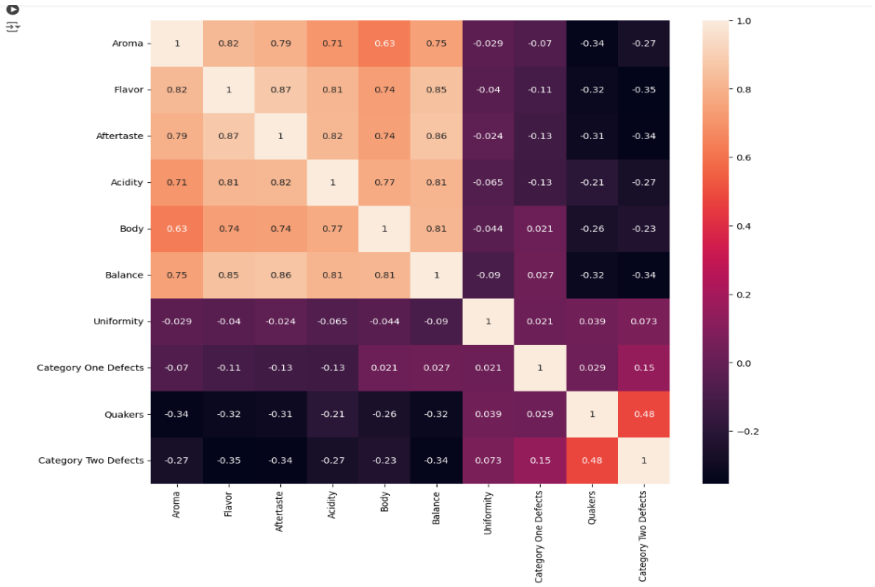
Data Collection and Preprocessing Phase

Date	9 July 2024
Team ID	team-739994
Project Title	Precise Coffee Quality Prediction
Maximum Marks	6 Marks

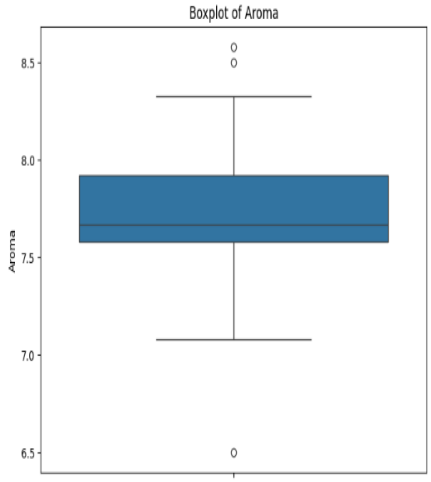
Data Exploration and Preprocessing Template

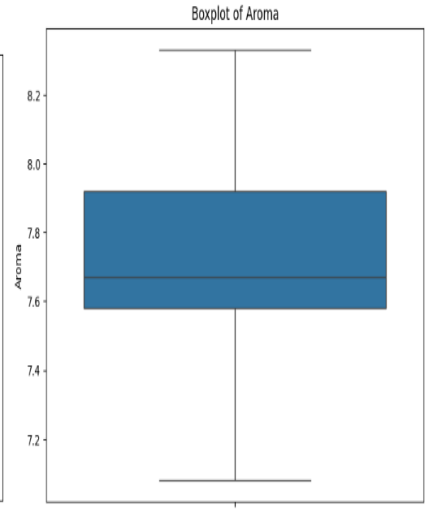
Dataset variables will be statistically analyzed to identify patterns and outliers, with python employed for preprocessing tasks like normalization and feature engineering .Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>Dimensions:
207 rows x 19 columns
<u>Descriptive Statistics:</u>
<table>
<tr><th></th><th>ID</th><th>Number of Bags</th><th>Aroma</th><th>Flavor</th><th>Aftertaste</th><th>Acidity</th><th>Body</th><th>Balance</th></tr>
<tr><td>count</td><td>207.000000</td><td>207.000000</td><td>207.000000</td><td>207.000000</td><td>207.000000</td><td>207.00000</td><td>207.000000</td><td>207.000000</td></tr>
<tr><td>mean</td><td>103.000000</td><td>155.449275</td><td>7.721063</td><td>7.744734</td><td>7.599758</td><td>7.69029</td><td>7.640918</td><td>7.644058</td></tr>
<tr><td>std</td><td>59.899917</td><td>244.484868</td><td>0.287626</td><td>0.279613</td><td>0.275911</td><td>0.25951</td><td>0.233499</td><td>0.256299</td></tr>
<tr><td>min</td><td>0.000000</td><td>1.000000</td><td>6.500000</td><td>6.750000</td><td>6.670000</td><td>6.83000</td><td>6.830000</td><td>6.670000</td></tr>
<tr><td>25%</td><td>51.500000</td><td>1.000000</td><td>7.580000</td><td>7.580000</td><td>7.420000</td><td>7.50000</td><td>7.500000</td><td>7.500000</td></tr>
<tr><td>50%</td><td>103.000000</td><td>14.000000</td><td>7.670000</td><td>7.750000</td><td>7.580000</td><td>7.67000</td><td>7.670000</td><td>7.670000</td></tr>
<tr><td>75%</td><td>154.500000</td><td>275.000000</td><td>7.920000</td><td>7.920000</td><td>7.750000</td><td>7.87500</td><td>7.750000</td><td>7.790000</td></tr>
<tr><td>max</td><td>206.000000</td><td>2240.000000</td><td>8.580000</td><td>8.500000</td><td>8.420000</td><td>8.58000</td><td>8.250000</td><td>8.420000</td></tr>
</table></td></tr>
<tr><td>Univariate Analysis</td><td> 
</td></tr>
</table>

Bivariate Analysis	  
Multivariate Analysis	

Outliers and Anomalies





Data Preprocessing Code Screenshots

Loading Data

```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    df = pd.read_csv("/content/beans_data.csv")
    df
```

	ID	Number of Bags	Bag Weight	Variety	Processing Method	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Overall	Total Cup Points	Moisture Percentage	Category One Defects	Quakers	Color	Category Two Defects
0	0	1	35 kg	Caetillo	Double Anaerobic Washed	8.58	8.50	8.42	8.58	8.25	8.42	10.0	8.58	89.33	11.8	0	0	green	3
1	1	1	80 kg	Gesha	Washed / Wet	8.50	8.50	7.92	8.00	7.92	8.25	10.0	8.50	87.58	10.5	0	0	blue-green	0
2	2	19	25 kg	Java	Semi Washed	8.33	8.42	8.08	8.17	7.92	8.17	10.0	8.33	87.42	10.4	0	0	yellowish	2
3	3	1	22 kg	Gesha	Washed / Wet	8.08	8.17	8.17	8.25	8.17	8.08	10.0	8.25	87.17	11.8	0	0	green	0
4	4	2	24 kg	Red Bourbon	Honey,Mossto	8.33	8.33	8.08	8.25	7.92	7.92	10.0	8.25	87.08	11.6	0	2	yellow-green	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
202	202	2240	60 kg	Mundo Novo	Natural / Dry	7.17	7.17	6.92	7.17	7.42	7.17	10.0	7.08	80.08	11.4	0	0	green	4
203	203	300	30 kg	SHG	Natural / Dry	7.33	7.08	6.75	7.17	7.42	7.17	10.0	7.08	80.00	10.4	0	2	green	12
204	204	343	60 kg	Catimor	Washed / Wet	7.25	7.17	7.08	7.00	7.08	7.08	10.0	7.00	79.67	11.6	0	9	green	11
205	205	1	2 kg	Maragogype	Natural / Dry	6.50	6.75	6.75	7.17	7.08	7.00	10.0	6.83	78.00	11.0	0	12	bluish-green	13
206	206	600	60 kg	Mundo Novo	SEMI-LAVADO	7.25	7.08	6.67	6.83	6.83	6.67	10.0	6.67	78.00	11.3	0	0	green	1

207 rows × 19 columns

Handling Missing Data

```
df.isnull().sum()
```

```
ID                        0
Number of Bags            0
Bag Weight                0
Variety                   6
Processing Method         5
Aroma                     0
Flavor                    0
Aftertaste                0
Acidity                   0
Body                      0
Balance                   0
Uniformity                0
Overall                   0
Total Cup Points          0
Moisture Percentage       0
Category One Defects      0
Quakers                   0
Color                     0
Category Two Defects      0
dtype: int64
```

```
[ ] df.dropna(inplace=True)
```

```
df.isna().sum()
```

```
ID                        0
Number of Bags            0
Bag Weight                0
Variety                   0
Processing Method         0
Aroma                     0
Flavor                    0
Aftertaste                0
Acidity                   0
Body                      0
Balance                   0
Uniformity                0
Overall                   0
Total Cup Points          0
Moisture Percentage       0
Category One Defects      0
Quakers                   0
Color                     0
Category Two Defects      0
dtype: int64
```

Data Transformation

```
[ ] from sklearn.preprocessing import LabelEncoder
    label_encoder = LabelEncoder()
    df1['Color_Encoded'] = label_encoder.fit_transform(df1['Color'])
    df1 = df1.drop(['Color'],axis=1)
```

```
df1
```

	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Category One Defects	Quakers	Category Two Defects	Color_Encoded
0	8.58	8.50	8.42	8.58	8.25	8.42	10.0	0	0	3	4
1	8.50	8.50	7.92	8.00	7.92	8.25	10.0	0	0	0	0
2	8.33	8.42	8.08	8.17	7.92	8.17	10.0	0	0	2	11
3	8.08	8.17	8.17	8.25	8.17	8.08	10.0	0	0	0	4
4	8.33	8.33	8.08	8.25	7.92	7.92	10.0	0	2	2	10
...	...	...	...	...	...	...	...	...	...	...	...
202	7.17	7.17	6.92	7.17	7.42	7.17	10.0	0	0	4	4
203	7.33	7.08	6.75	7.17	7.42	7.17	10.0	0	2	12	4
204	7.25	7.17	7.08	7.00	7.08	7.08	10.0	0	9	11	4

Feature Engineering

205	6.50	6.75	6.75	7.17	7.08	7.00	10.0	0	12	13	1
206	7.25	7.08	6.67	6.83	6.83	6.67	10.0	0	0	1	4

197 rows × 11 columns

```
df1['Bean_Status']='Healthy'
condition_healthy=(df1['Category One Defects']==0) & (df1['Category Two Defects']==0)
df1.loc[condition_healthy,'Bean_Status']='Healthy'
condition_unhealthy=(df1['Category One Defects']!=0) & (df1['Category Two Defects']!=0)
df1.loc[condition_unhealthy,'Bean_Status']='Unhealthy'
```

```
[ ] df1
```

	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Category One Defects	Quakers	Category Two Defects	Color_Encoded	Bean_Status
0	8.58	8.50	8.42	8.58	8.25	8.42	10.0	0	0	3	4	Healthy
1	8.50	8.50	7.92	8.00	7.92	8.25	10.0	0	0	0	0	Healthy
2	8.33	8.42	8.08	8.17	7.92	8.17	10.0	0	0	2	11	Healthy
3	8.08	8.17	8.17	8.25	8.17	8.08	10.0	0	0	0	4	Healthy

Save Processed Data

```
[ ] import pickle
    import warnings
```

```
[ ] with open("./coffee_quality_prediction(rfc).pkl","wb") as f:
      pickle.dump(RFC,f)
```