Data Collection and Preprocessing Phase

Date	7 June 2024
Team ID	SWTID1720112707
Project Title	Anemia Sense: Leveraging Machine Learning For Precise Anemia Recognitions
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	`df.head()`

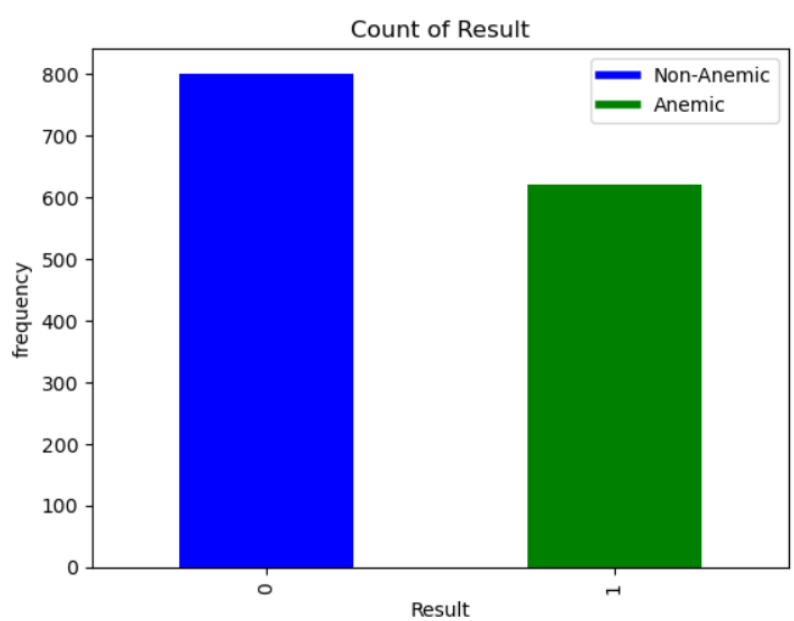
	Gender	Hemoglobin	MCH	MCHC	MCV	Result
0	1	14.9	22.7	29.1	83.7	0
1	0	15.9	25.4	28.3	72.0	0
2	0	9.0	21.5	29.6	71.2	1
3	0	14.9	16.0	31.4	87.5	0
4	1	14.7	22.0	28.2	99.5	0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1421 entries, 0 to 1420
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Gender      1421 non-null   int64
 1   Hemoglobin  1421 non-null   float64
 2   MCH         1421 non-null   float64
 3   MCHC        1421 non-null   float64
 4   MCV         1421 non-null   float64
 5   Result      1421 non-null   int64
dtypes: float64(4), int64(2)
memory usage: 66.7 KB
```

```
df.shape
```

```
(1421, 6)
```

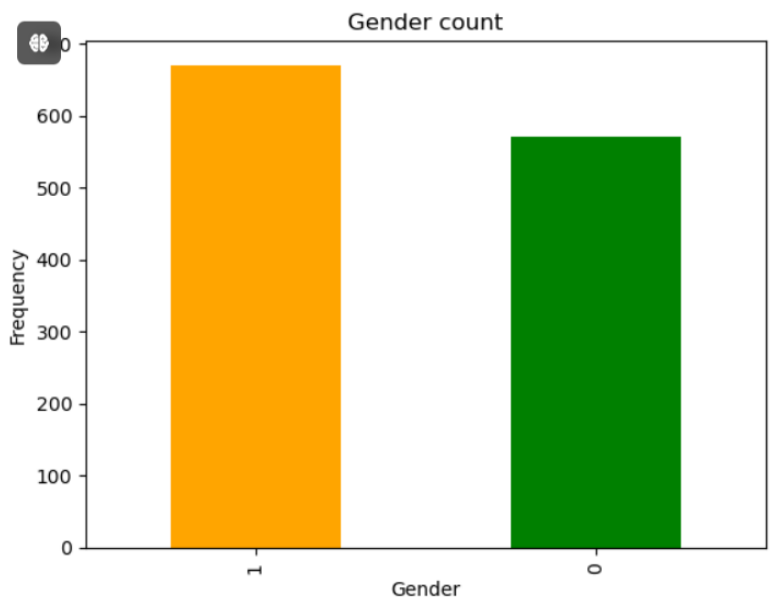


```
df.describe()
```

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
count	1240.000000	1240.000000	1240.000000	1240.000000	1240.000000	1240.000000
mean	0.540323	13.218145	22.903952	30.277984	85.620968	0.500000
std	0.498573	1.976190	3.993624	1.394515	9.673794	0.500202
min	0.000000	6.600000	16.000000	27.800000	69.400000	0.000000
25%	0.000000	11.500000	19.400000	29.100000	77.300000	0.000000
50%	1.000000	13.000000	22.700000	30.400000	85.300000	0.500000
75%	1.000000	14.900000	26.200000	31.500000	94.225000	1.000000
max	1.000000	16.900000	30.000000	32.500000	101.600000	1.000000

Univariate Analysis

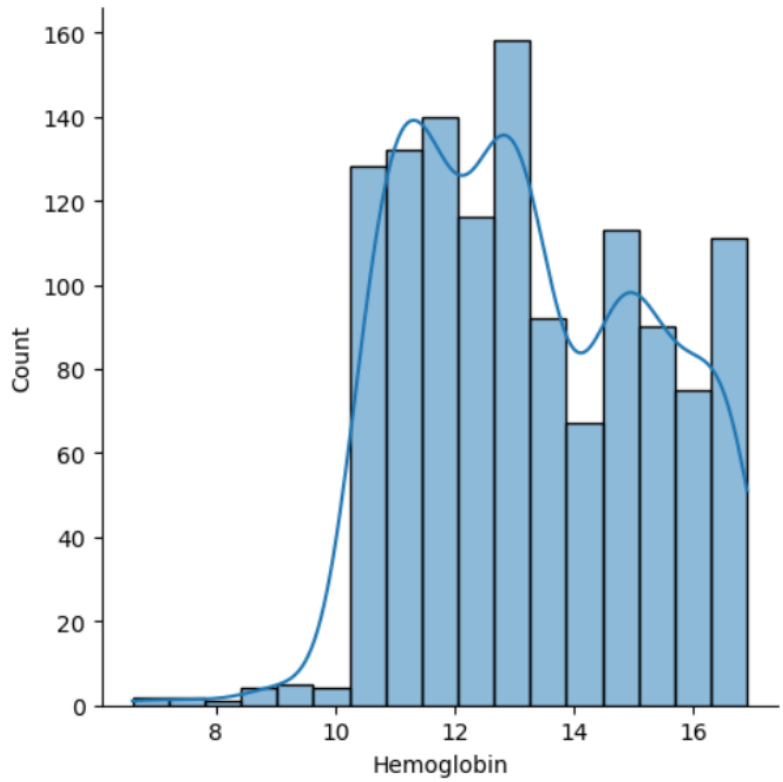
```
# Univariate Analysis

output = df['Gender'].value_counts()
output.plot(kind = 'bar', color=['orange', 'green'])
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.title('Gender count')
plt.show()
```



```
sns.displot(df['Hemoglobin'], kde = True)
```

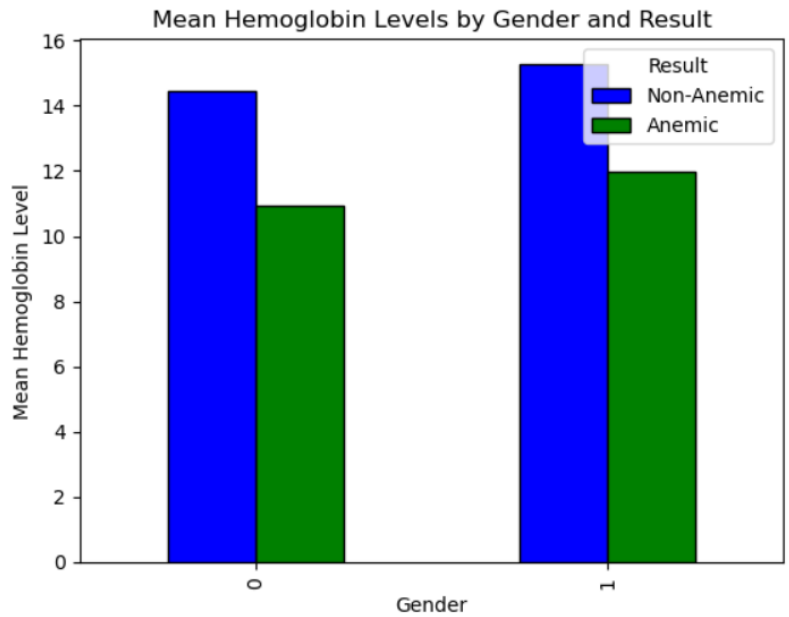
```
D:\Anaconda\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning
ert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
<seaborn.axisgrid.FacetGrid at 0x24dbfc157d0>
```

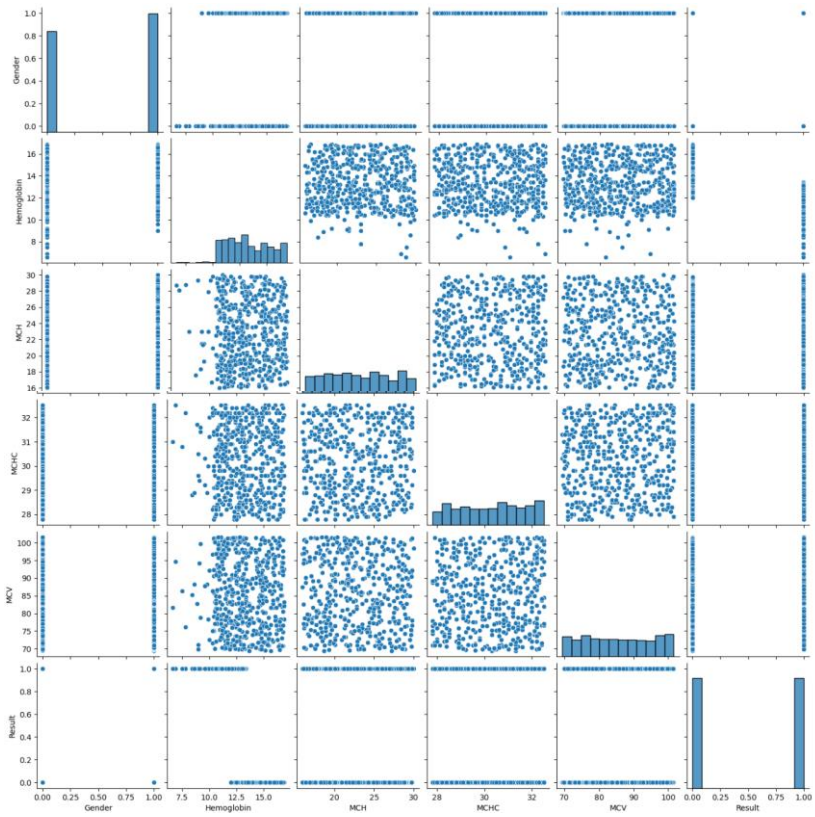


Bivariate Analysis

```
# Bivariate Analysis

# Calculate mean hemoglobin levels grouped by gender and result
mean_hemoglobin = df.groupby(['Gender', 'Result'])['Hemoglobin'].mean().reset_index()

# Pivot the data to get a format suitable for plotting
mean_hemoglobin_pivot = mean_hemoglobin.pivot(index='Gender', columns='Result', values='Hemoglobin')

# Plot the histogram
mean_hemoglobin_pivot.plot(kind='bar', color=['blue', 'green'], edgecolor='black')
plt.xlabel('Gender')
plt.ylabel('Mean Hemoglobin Level')
plt.title('Mean Hemoglobin Levels by Gender and Result')
plt.legend(title='Result', labels=['Non-Anemic', 'Anemic'])
plt.show()
```

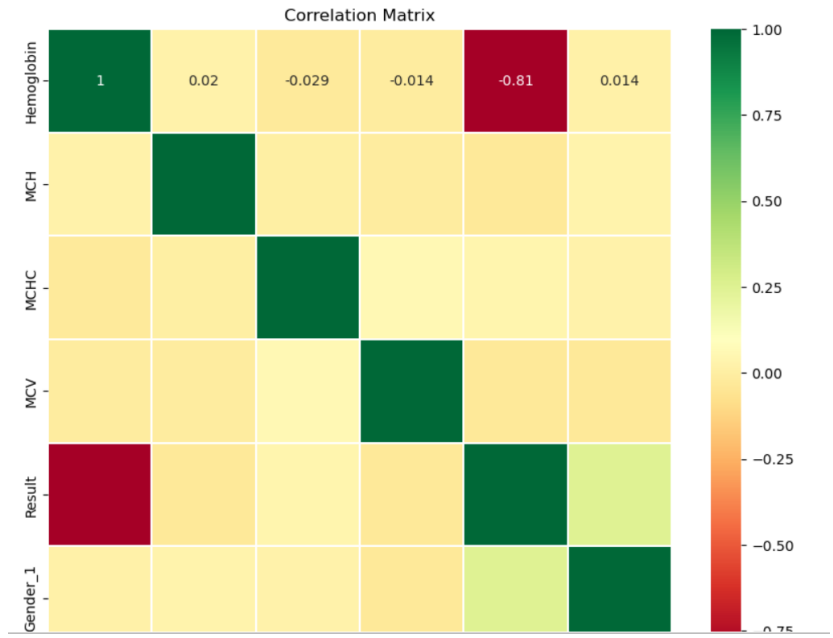
	

Multivariate Analysis	```
	# Multivariate Analysis
	
	# Convert infinite values to NaN
	df.replace([np.inf, -np.inf], np.nan, inplace=True)
	
	df.dropna(inplace=True)
	
	sns.pairplot(df)
	plt.show()
	``




```python
# Multivariate Analysis

print(df.dtypes)

df = df.apply(pd.to_numeric, errors='coerce')

print(df.isnull().sum())
df = df.dropna()  # or use an appropriate imputation method

df = pd.get_dummies(df, columns=['Gender'], drop_first=True)

plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='RdYlGn', linewidths=0.2)
plt.title('Correlation Matrix')
plt.show()
```

```
Gender          int64
Hemoglobin    float64
MCH           float64
MCHC          float64
MCV           float64
Result          int64
dtype: object
Gender        0
Hemoglobin    0
MCH           0
MCHC          0
MCV           0
Result        0
dtype: int64
```



Data Preprocessing Code Screenshots

Loading Data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('D:/anemia.csv')
```

Handling Missing Data

```
print("Initial Count of Missing Values in the Dataset\n")
print(df.isnull().sum())

# Handling missing values
# Filling missing numerical values with the median
for column in df.select_dtypes(include=[np.number]).columns:
    df[column].fillna(df[column].median(), inplace=True)

for column in df.select_dtypes(include=[object]).columns:
    df[column].fillna(df[column].mode()[0], inplace=True)

print("\nFinal Count of Missing Values in the Dataset\n")
print(df.isnull().sum())
```

Data Transformation	 #we can see that the female count is more than the male so, # we can balance it using the undersampling from sklearn.utils import resample majorclass = df[df['Result'] == 0] minorclass = df[df['Result'] == 1] major_downsample = resample(majorclass, replace=False, n_samples=len(minorclass), random_state=42) df = pd.concat([major_downsample, minorclass]) print(df['Result'].value_counts())

Feature Engineering	
	for column in df.select_dtypes(include=[np.number]).columns:
	df[column].fillna(df[column].median(), inplace=True)
	
	for column in df.select_dtypes(include=[object]).columns:
	df[column].fillna(df[column].mode()[0], inplace=True)

Save Processed Data	 df = pd.concat([major_downsample, minorclass])