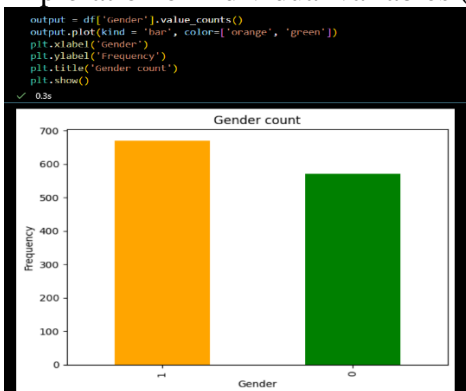


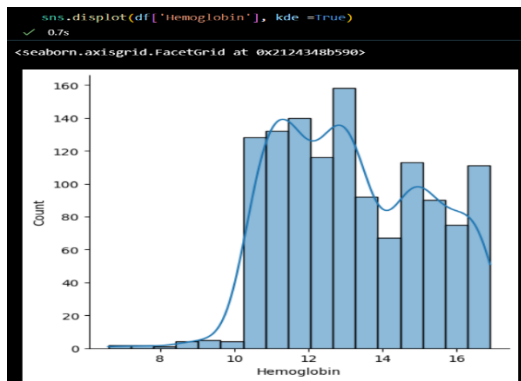
Data Collection and Preprocessing Phase

Date	9 JULY 2024
Team ID	739661
Project Title	Anemiasense: Leveraging Machine Learning For Precise Anemia Recognitions
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

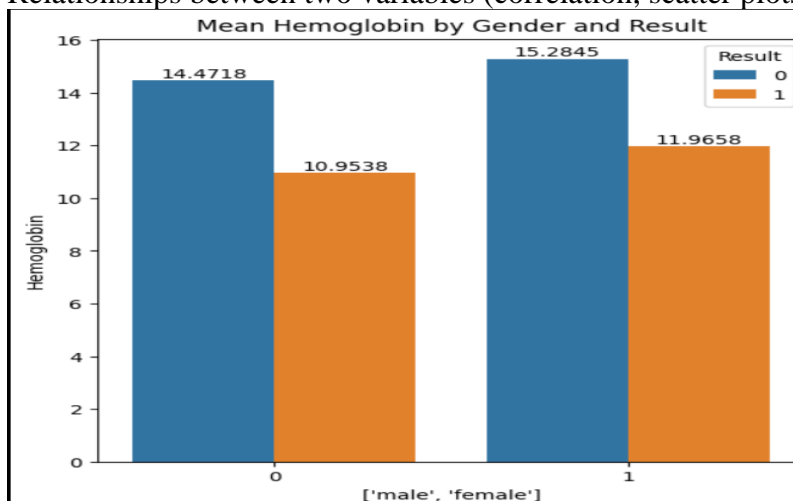
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	Basic statistics, dimensions, and structure of the data.
Univariate Analysis	<p>Exploration of individual variables (mean, median, mode, etc.).</p>  <pre> output = df['Gender'].value_counts() output.plot(kind = 'bar', color=['orange', 'green']) plt.xlabel('Gender') plt.ylabel('frequency') plt.title('Gender count') plt.show() </pre>



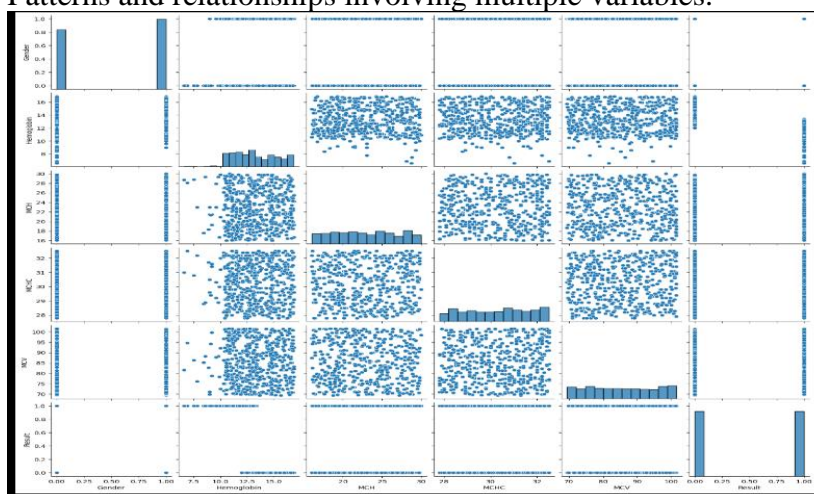
Bivariate Analysis

Relationships between two variables (correlation, scatter plots).



Multivariate Analysis

Patterns and relationships involving multiple variables.



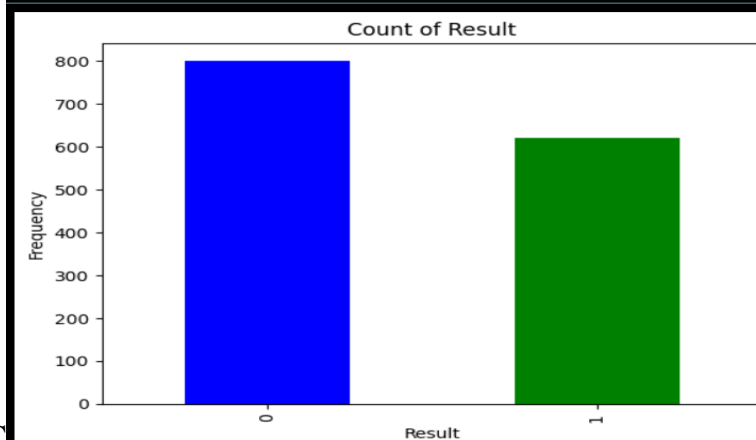
	<pre>sns.heatmap(df.corr(),annot=True,cmap='RdYlGn',linewidths=0.2) #data.corr()-->correlation matrix fig=plt.gcf() fig.set_size_inches(10,8) plt.show()</pre> 																																										
Data Preprocessing Code Screenshots:	<pre>#importing the libraries import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns</pre>																																										
Loading Data	<p>Code to load the dataset into the preferred environment</p> <pre>df = pd.read_csv('anemia.csv') df.head()</pre> <table><thead><tr><th></th><th>Gender</th><th>Hemoglobin</th><th>MCH</th><th>MCHC</th><th>MCV</th><th>Result</th></tr></thead><tbody><tr><td>0</td><td>1</td><td>14.9</td><td>22.7</td><td>29.1</td><td>83.7</td><td>0</td></tr><tr><td>1</td><td>0</td><td>15.9</td><td>25.4</td><td>28.3</td><td>72.0</td><td>0</td></tr><tr><td>2</td><td>0</td><td>9.0</td><td>21.5</td><td>29.6</td><td>71.2</td><td>1</td></tr><tr><td>3</td><td>0</td><td>14.9</td><td>16.0</td><td>31.4</td><td>87.5</td><td>0</td></tr><tr><td>4</td><td>1</td><td>14.7</td><td>22.0</td><td>28.2</td><td>99.5</td><td>0</td></tr></tbody></table>		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
	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																																					
Handling Missing Data	<p>Code for identifying and handling missing values.</p> <pre>#checking for null values df.isnull().sum()</pre> <table><tbody><tr><td>Gender</td><td>0</td></tr><tr><td>Hemoglobin</td><td>0</td></tr><tr><td>MCH</td><td>0</td></tr><tr><td>MCHC</td><td>0</td></tr><tr><td>MCV</td><td>0</td></tr><tr><td>Result</td><td>0</td></tr></tbody></table> <pre>dtype: int64</pre>	Gender	0	Hemoglobin	0	MCH	0	MCHC	0	MCV	0	Result	0																														
Gender	0																																										
Hemoglobin	0																																										
MCH	0																																										
MCHC	0																																										
MCV	0																																										
Result	0																																										

Data Transformation

```
#0-not anemic 1-anemic
#checking for the count of anemia and not anemia

results = df['Result'].value_counts()
results.plot(kind = 'bar', color=['blue', 'green'])
plt.xlabel('Result')
plt.ylabel('Frequency')
plt.title('Count of Result')
plt.show()
```

✓ 0.6s



```
#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])

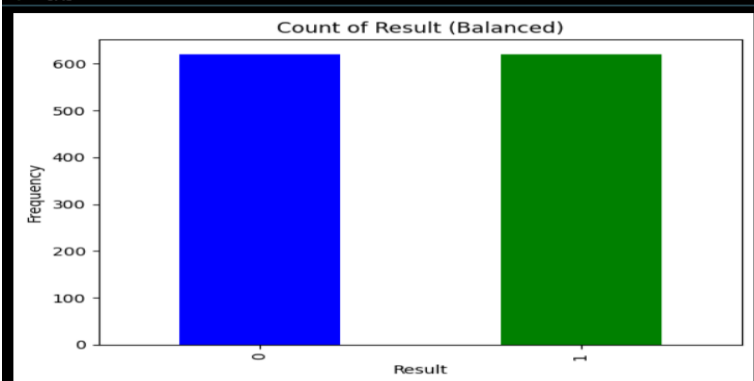
print(df['Result'].value_counts())
```

✓ 0.3s Python

```
Result
0    620
1    620
Name: count, dtype: int64

# Plot the balanced gender counts
result_balanced = df['Result'].value_counts()
result_balanced.plot(kind='bar', color=['blue', 'green'])
plt.xlabel('Result')
plt.ylabel('Frequency')
plt.title('Count of Result (Balanced)')
plt.show()
```

✓ 0.4s



Feature Engineering

Code for creating new features or modifying existing ones.

	<pre>x = df.drop('Result', axis = 1) x ✓ 0.0s</pre> <table><thead><tr><th></th><th>Gender</th><th>Hemoglobin</th><th>MCH</th><th>MCHC</th><th>MCV</th></tr></thead><tbody><tr><td>1234</td><td>1</td><td>16.6</td><td>18.8</td><td>28.1</td><td>70.9</td></tr><tr><td>1188</td><td>0</td><td>15.3</td><td>18.3</td><td>30.4</td><td>93.4</td></tr><tr><td>106</td><td>0</td><td>14.8</td><td>20.4</td><td>28.5</td><td>91.1</td></tr><tr><td>954</td><td>0</td><td>14.6</td><td>16.9</td><td>31.9</td><td>78.1</td></tr><tr><td>112</td><td>0</td><td>15.9</td><td>28.7</td><td>31.0</td><td>81.6</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>1415</td><td>1</td><td>13.2</td><td>20.1</td><td>28.8</td><td>91.2</td></tr><tr><td>1416</td><td>0</td><td>10.6</td><td>25.4</td><td>28.2</td><td>82.9</td></tr><tr><td>1417</td><td>1</td><td>12.1</td><td>28.3</td><td>30.4</td><td>86.9</td></tr><tr><td>1418</td><td>1</td><td>13.1</td><td>17.7</td><td>28.1</td><td>80.7</td></tr><tr><td>1420</td><td>0</td><td>11.8</td><td>21.2</td><td>28.4</td><td>98.1</td></tr></tbody></table> <pre>Y = df['Result'] Y ✓ 0.0s</pre> <table><tbody><tr><td>1234</td><td>0</td></tr><tr><td>1188</td><td>0</td></tr><tr><td>106</td><td>0</td></tr><tr><td>954</td><td>0</td></tr><tr><td>112</td><td>0</td></tr><tr><td>...</td><td>..</td></tr><tr><td>1415</td><td>1</td></tr><tr><td>1416</td><td>1</td></tr><tr><td>1417</td><td>1</td></tr><tr><td>1418</td><td>1</td></tr><tr><td>1420</td><td>1</td></tr></tbody></table> <p>Name: Result, Length: 1240, dtype: int64</p>		Gender	Hemoglobin	MCH	MCHC	MCV	1234	1	16.6	18.8	28.1	70.9	1188	0	15.3	18.3	30.4	93.4	106	0	14.8	20.4	28.5	91.1	954	0	14.6	16.9	31.9	78.1	112	0	15.9	28.7	31.0	81.6	1415	1	13.2	20.1	28.8	91.2	1416	0	10.6	25.4	28.2	82.9	1417	1	12.1	28.3	30.4	86.9	1418	1	13.1	17.7	28.1	80.7	1420	0	11.8	21.2	28.4	98.1	1234	0	1188	0	106	0	954	0	112	0	1415	1	1416	1	1417	1	1418	1	1420	1
	Gender	Hemoglobin	MCH	MCHC	MCV																																																																																										
1234	1	16.6	18.8	28.1	70.9																																																																																										
1188	0	15.3	18.3	30.4	93.4																																																																																										
106	0	14.8	20.4	28.5	91.1																																																																																										
954	0	14.6	16.9	31.9	78.1																																																																																										
112	0	15.9	28.7	31.0	81.6																																																																																										
...																																																																																										
1415	1	13.2	20.1	28.8	91.2																																																																																										
1416	0	10.6	25.4	28.2	82.9																																																																																										
1417	1	12.1	28.3	30.4	86.9																																																																																										
1418	1	13.1	17.7	28.1	80.7																																																																																										
1420	0	11.8	21.2	28.4	98.1																																																																																										
1234	0																																																																																														
1188	0																																																																																														
106	0																																																																																														
954	0																																																																																														
112	0																																																																																														
...	..																																																																																														
1415	1																																																																																														
1416	1																																																																																														
1417	1																																																																																														
1418	1																																																																																														
1420	1																																																																																														
Save Processed Data	<p>Code to save the cleaned and processed data for future use.</p> <pre>from sklearn.model_selection import train_test_split ✓ 0.2s</pre> <pre>x_train, x_test, y_train, y_test = train_test_split(X, Y , test_size=0.2, random_state=20) ✓ 0.0s</pre> <pre>print(x_train.shape) print(x_test.shape) print(y_train.shape) print(y_test.shape) ✓ 0.0s</pre> <pre>(992, 5) (248, 5) (992,) (248,)</pre>																																																																																														