Data Collection and Preprocessing Phase

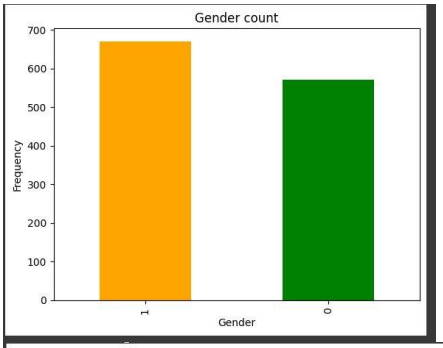
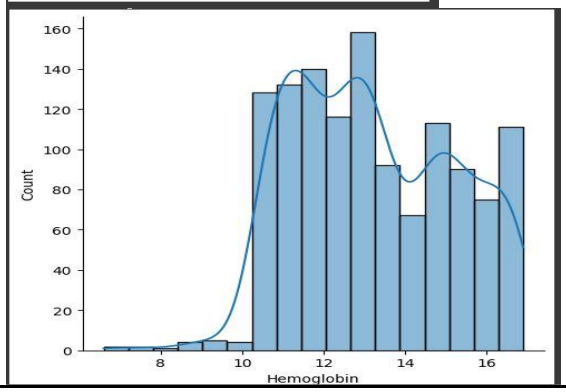
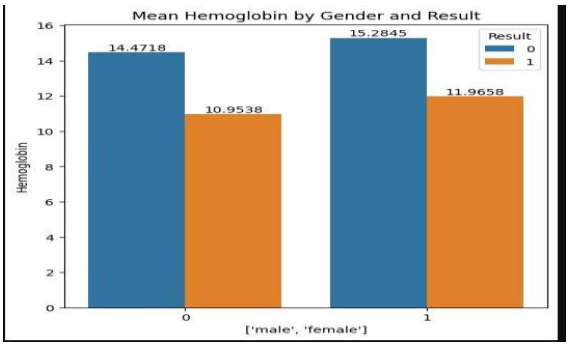
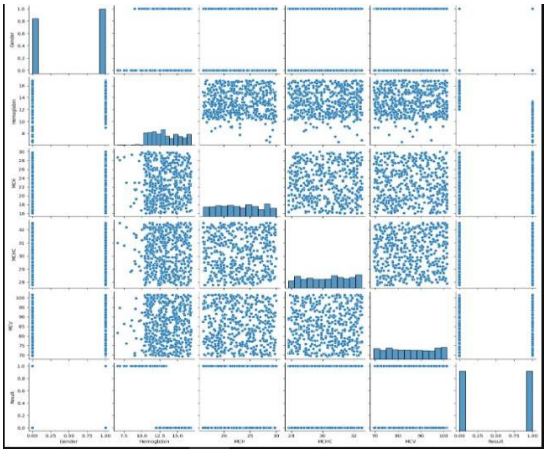
Date	1 August 2025
Skillwallet ID	SWUID20250194750
Project Title	Anemia Sense: Leveraging Machine Learning For Precise Anemia
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employedforpreprocessingtaskslikenormalizationandfeatureengineering.Datacleaningwill address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

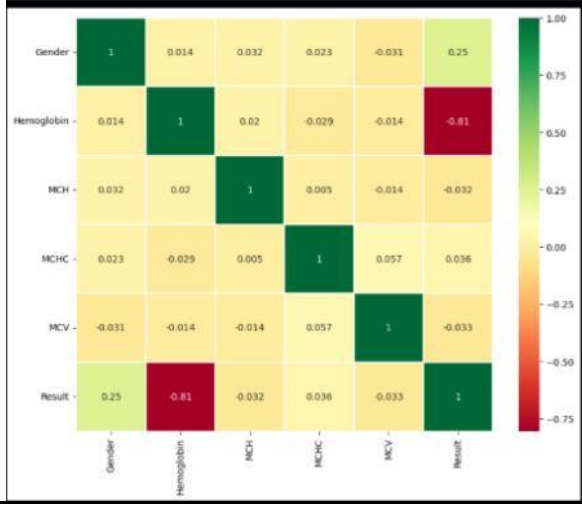
Section	Description
Data Overview	Dimension: 614rows×13columns Descriptive statistics: (see table below)
Univariate Analysis	

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
count	1421.000000	1421.000000	1421.000000	1421.000000	1421.000000	1421.000000
mean	0.520760	13.412738	22.905630	30.251232	85.523786	0.436312
std	0.499745	1.974546	3.969375	1.400898	9.636701	0.496102
min	0.000000	6.600000	16.000000	27.800000	69.400000	0.000000
25%	0.000000	11.700000	19.400000	29.000000	77.300000	0.000000
50%	1.000000	13.200000	22.700000	30.400000	85.300000	0.000000
75%	1.000000	15.000000	26.200000	31.400000	94.200000	1.000000
max	1.000000	16.900000	30.000000	32.500000	101.600000	1.000000

	 
Bivariate Analysis	
Multivariate Analysis	

```
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()
```

✓ 0.7s



Splitting Data into Train and Test

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,Y, test_size=0.2, random_state=20)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(992, 5)
(248, 5)
(992,)
(248,)
```

Data Preprocessing Code Screenshots

Loading Data

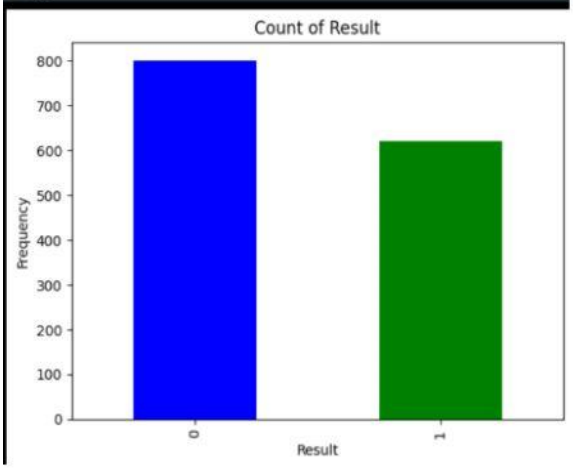
```
df = pd.read_csv('anemia.csv')
df
```

	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
...	...	...	...	...	...	...
1416	0	10.6	25.4	28.2	82.9	1
1417	1	12.1	28.3	30.4	86.9	1
1418	1	13.1	17.7	28.1	80.7	1
1419	0	14.3	16.2	29.5	95.2	0
1420	0	11.8	21.2	28.4	98.1	1

1421 rows × 6 columns

Handling Missing Data	```python
	df = pd.read_csv('anemia.csv')
	
	df.info()
	df.shape
	``
	``
	<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
	(1421, 6)
	``
	``python
	df.isnull().sum()
	``
	|
		0

Gender	0
Hemoglobin	0
MCH	0
MCHC	0
MCV	0
Result	0 | dtype: int64

Handling Imbalanced Values	```python
	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
	``
	 
	
	``python
	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())
	``
	``
	Result
	0 620
	1 620
	Name: count, dtype: int64
	``

```
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