

## Data Collection and Preprocessing Phase

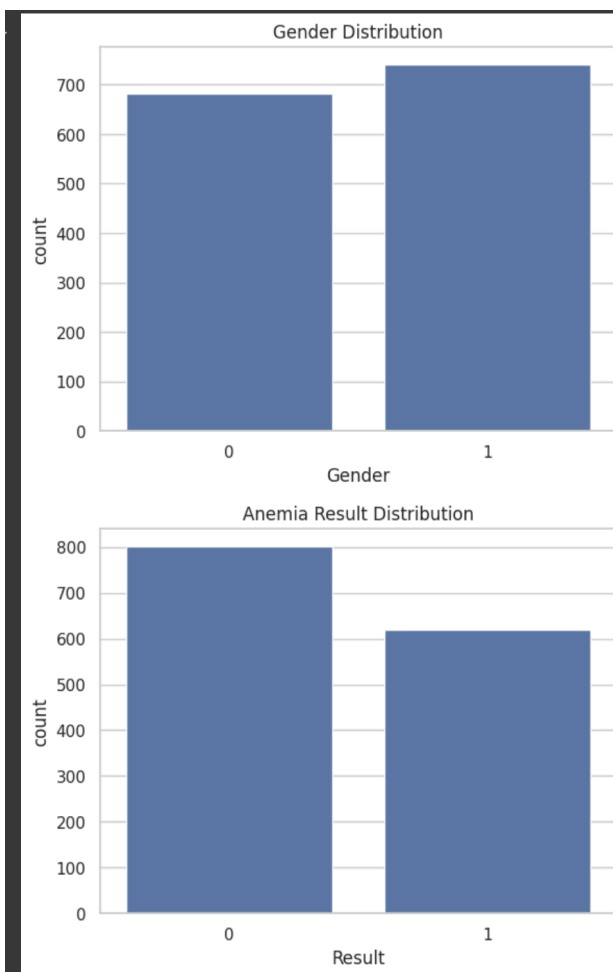
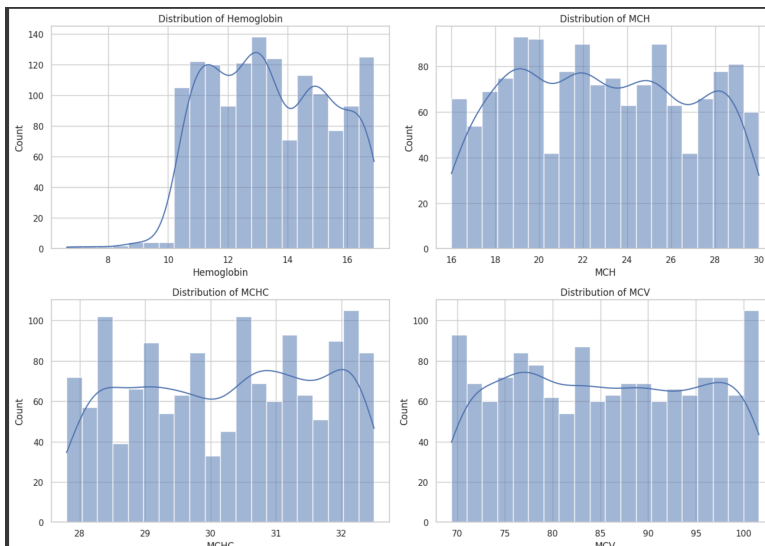
Date	14 JUNE 2025
Team ID	SWTID1749621188
Project Title	Anemia Sense Leveraging-Machine Learning-For-Precise Anemia Recognition
Maximum Marks	6 Marks

### Data Exploration and Preprocessing Template

The dataset was obtained from a structured CSV file containing hematological parameters like Hemoglobin, MCH, MCHC, MCV, and a binary anemia result. Initial exploration involved checking the shape, data types, and value distributions. No missing values or duplicates were found. Numerical columns were analyzed through descriptive statistics and visualized using histograms, KDE plots, and box plots. Outliers were detected in Hemoglobin and MCH and reviewed for impact. Since all features were numerical, no encoding was required. The cleaned and processed data was then prepared for further modeling.

Section	Description
Data Overview	<p><b>DIMENSIONS:</b></p> <p>Shape of the dataset (rows, columns): (1421, 6)</p> <p><b>DESCRIPTIVE STATISTICS:</b></p> <pre> Descriptive statistics for numerical columns:       Gender  Hemoglobin  MCH  MCHC  MCV  \ count  1421.000000  1421.000000  1421.000000  1421.000000  1421.000000 mean    0.520760    13.412738    22.905630    30.251232    85.523786 std     0.499745     1.974546     3.969375     1.400898     9.636701 min     0.000000     6.600000    16.000000    27.800000    69.400000 25%     0.000000    11.700000    19.400000    29.000000    77.300000 50%     1.000000    13.200000    22.700000    30.400000    85.300000 75%     1.000000    15.000000    26.200000    31.400000    94.200000 max     1.000000    16.900000    30.000000    32.500000   101.600000        Result count  1421.000000 mean    0.436312 std     0.496102 min     0.000000 25%     0.000000 50%     0.000000 75%     1.000000 max     1.000000           </pre>

## Univariate Analysis

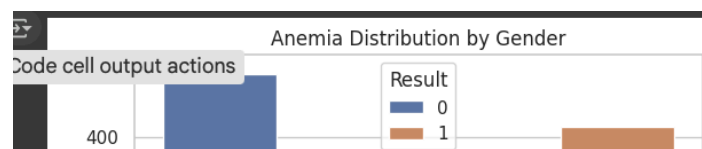
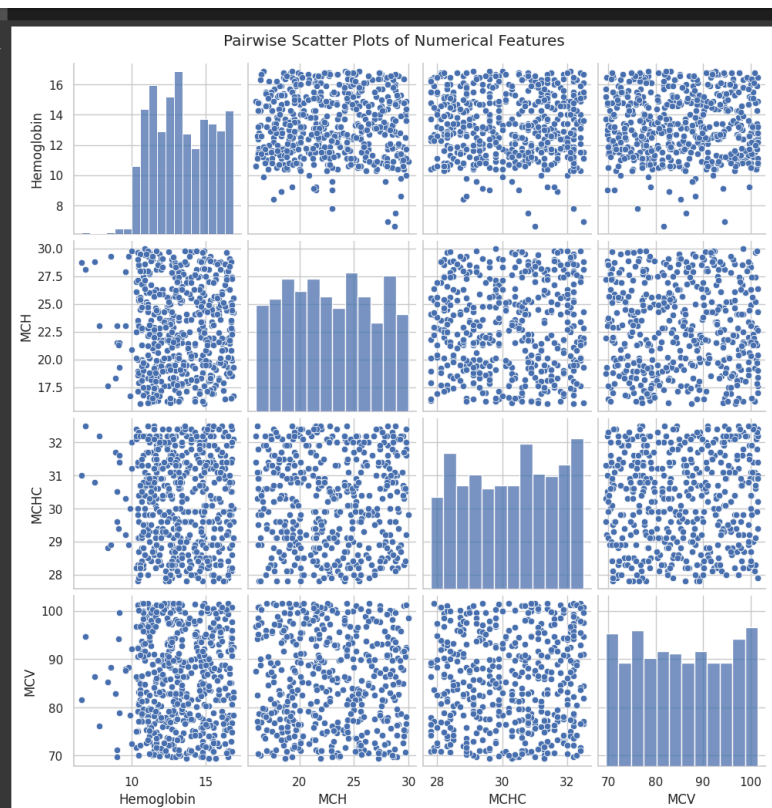
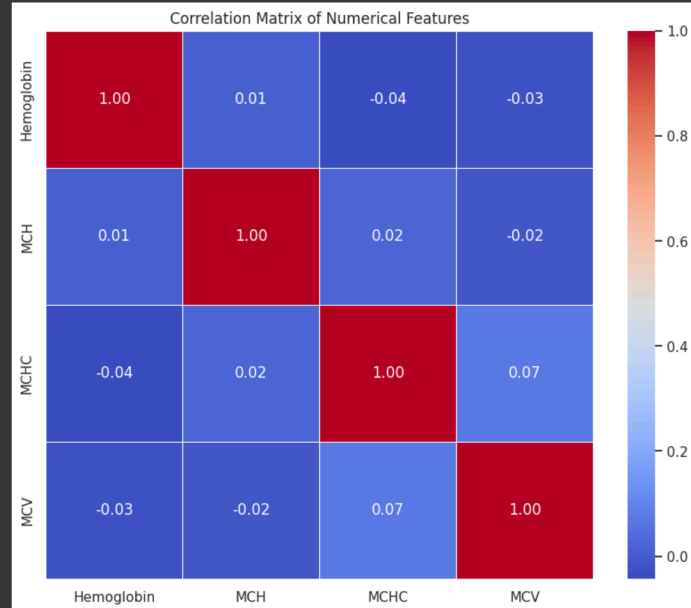


## Bivariate Analysis

A. Numerical vs. Numerical:

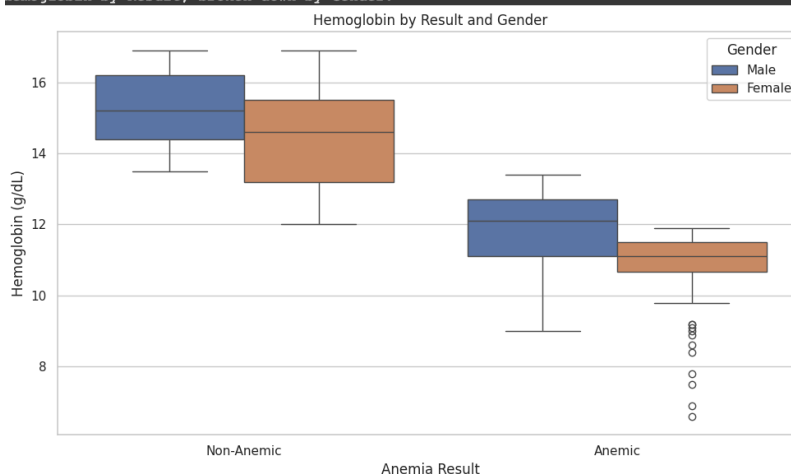
Correlation Matrix:

	Hemoglobin	MCH	MCHC	MCV
Hemoglobin	1.000000	0.014081	-0.042597	-0.025885
MCH	0.014081	1.000000	0.018795	-0.015948
MCHC	-0.042597	0.018795	1.000000	0.068450
MCV	-0.025885	-0.015948	0.068450	1.000000

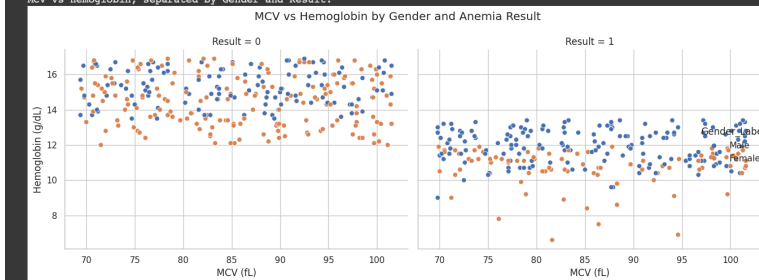


## Multivariate Analysis

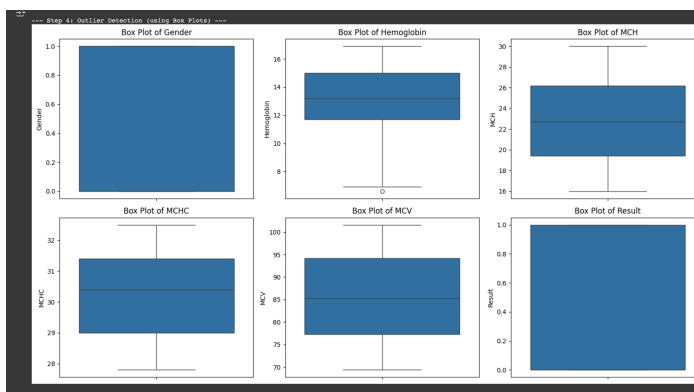
Hemoglobin by Result, broken down by Gender:



MCV vs Hemoglobin, separated by Gender and Result:



## Outliers and Anomalies



## Data Preprocessing Code Screenshots

## Loading Data

```
import pandas as pd

df = pd.read_csv('/content/Sheet 1-anemia.csv')
print(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
```

## Handling Missing Data

### 3) MISSING VALUE ANALYSIS

```
[ ] # Check for missing values
print("\nMissing values per column:")
missing_values = df.isnull().sum()
print(missing_values)
```

```
Missing values per column:
Gender      0
Hemoglobin  0
MCH         0
MCHC        0
MCV         0
Result      0
dtype: int64
```

```
[ ] # Calculate percentage of missing values
print("\nPercentage of missing values per column:")
missing_percentage = (df.isnull().sum() / len(df)) * 100
print(missing_percentage)
```

```
Percentage of missing values per column:
Gender      0.0
Hemoglobin  0.0
MCH         0.0
MCHC        0.0
MCV         0.0
Result      0.0
dtype: float64
```

```
[ ] # This dataset is quite clean with no missing values.
# If there were missing values, you would see output from the above commands.
```

## Data Transformation

### DATA TRANSFORMATION

```
# First, load the dataset
import pandas as pd
df = pd.read_csv('/content/Sheet 1-anemia.csv')

# Rename Gender for clarity
df['Gender_Label'] = df['Gender'].map({0: 'Female', 1: 'Male'})

# Optional: Normalize/scale numerical columns
from sklearn.preprocessing import StandardScaler

# Selecting numerical columns
num_cols = ['Hemoglobin', 'MCH', 'MCHC', 'MCV']
scaler = StandardScaler()
df[num_cols] = scaler.fit_transform(df[num_cols])
```

Feature Engineering	Attached the feature engineering code in the final submission
Save Processed Data	<pre>y_test.shape[0] # Save cleaned dataset df.to_csv('processed_anemia_data.csv', index=False)</pre>