
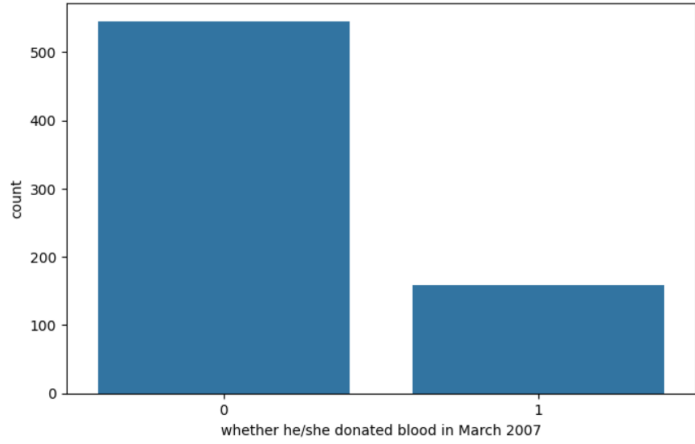


## Data Collection and Preprocessing Phase

Date	14 July 2024
Team ID	739963
Project Title	Blood Donation Prediction
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	 <p>The screenshot shows a Jupyter Notebook interface with the title 'EXPLORATORY DATA ANALYSIS'. The first cell contains the command <code>df.describe()</code>, which has been executed. The output is a summary statistics table for the dataset. The columns are: Recency (months), Frequency (times), Monetary (c.c. blood), Time (months), and whether he/she donated blood in March 2007. The rows show count, mean, std, min, 25%, 50%, 75%, and max for each column.</p>
Univariate Analysis	<p>Code + Text</p> <p><b>UNIVARIATE ANALYSIS</b></p> <pre>sns.countplot(x='whether he/she donated blood in March 2007',data=df)</pre> <p>&lt;Axes: xlabel='whether he/she donated blood in March 2007', ylabel='count'&gt;</p>  <p>The bar chart displays the distribution of blood donations. The x-axis is labeled 'whether he/she donated blood in March 2007' with values 0 and 1. The y-axis is labeled 'count' and ranges from 0 to 500. The bar for '0' (did not donate) is significantly higher, reaching approximately 550, while the bar for '1' (donated) is much lower, around 150.</p>

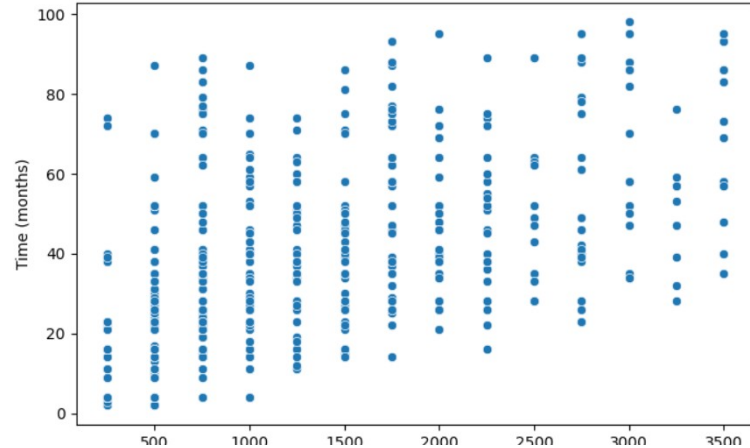
## Bivariate Analysis

CODE + TEXT

### BIVARIATE ANALYSIS

```
sns.scatterplot(x=df['Monetary (c.c. blood)'],y=df['Time (months)'])
```

<Axes: xlabel='Monetary (c.c. blood)', ylabel='Time (months)'

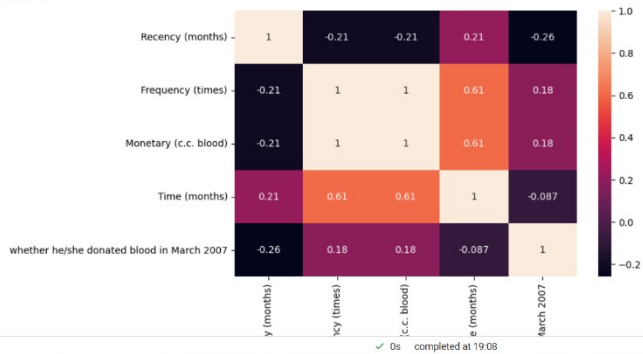


## Multivariate Analysis

### MULTIVARIATE ANALYSIS

```
[92] sns.heatmap(df.corr(),annot=True)
```

<Axes: >



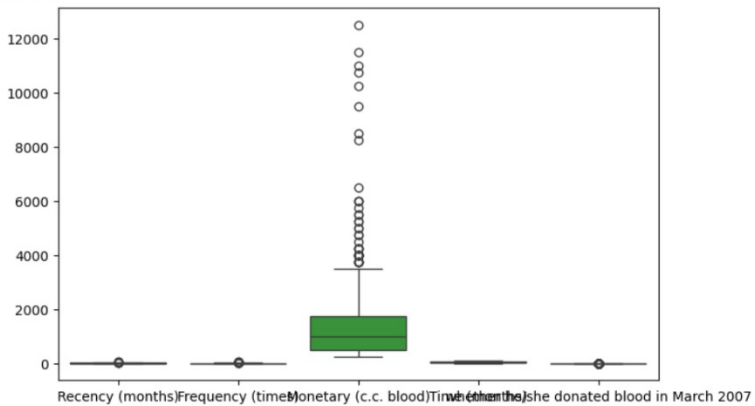
0s completed at 19:08

## Outliers and Anomalies

CODE + TEXT

```
sns.boxplot(df)
```

<Axes: >



## Data Preprocessing Code Screenshots

### Loading Data

```
[62]: df = pd.read_csv("/content/transfusion (2).csv")
df.head()
```

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)	whether he/she donated blood in March 2007
0	2	50	12500	98	1
1	0	13	3250	28	1
2	1	16	4000	35	1
3	2	20	5000	45	1
4	1	24	6000	77	0

### Handling Missing Data

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

```
Recency      0
Frequency    0
Monetary     0
Time         0
Donated      0
dtype: int64
```

### Data Transformation

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_features = scaler.fit_transform(df.drop(columns=["whether he/she donated blood in March 2007"]))
scaled_df = pd.DataFrame(scaled_features, columns=df.columns[:-1])

scaled_df["whether he/she donated blood in March 2007"] = df["whether he/she donated blood in March 2007"].values

# Display the transformed DataFrame
print("\nTransformed DataFrame:")
print(scaled_df.head())
```

```
Transformed DataFrame:
   Recency (months)  Frequency (times)  Monetary (c.c. blood)  Time (months) \
0         0.000000         0.923077         0.923077         0.270833
1         0.054954         0.230769         0.230769         0.020833
2         0.027027         0.461538         0.461538         0.125000
3         0.013514         0.846154         0.846154         0.343750
4         0.027027         0.615385         0.615385         0.208333

   whether he/she donated blood in March 2007
0                        1
1                        0
2                        1
3                        0
4                        1
```

### Feature Engineering

Attached the codes in final submission.

### Save Processed Data

-