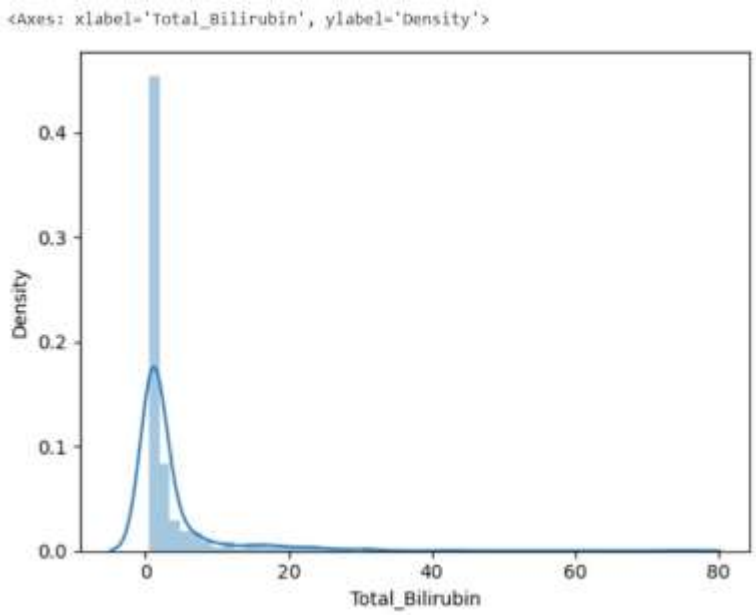
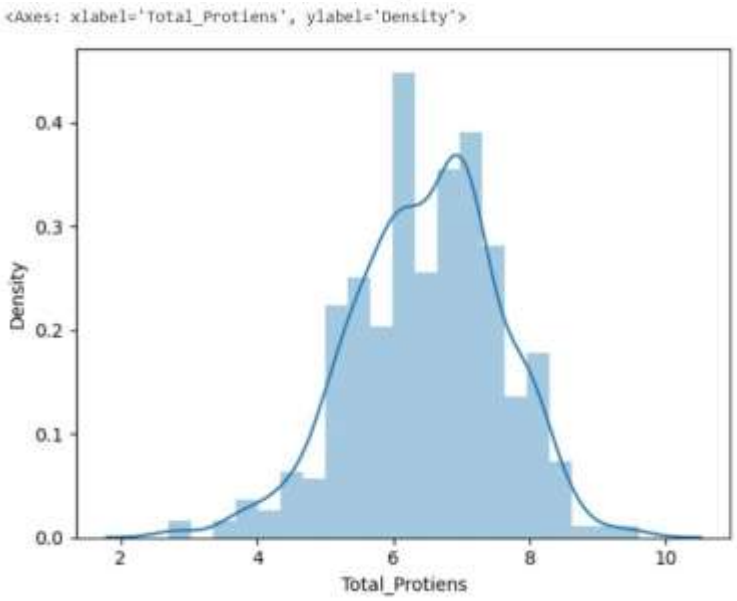
Data Collection and Preprocessing Phase

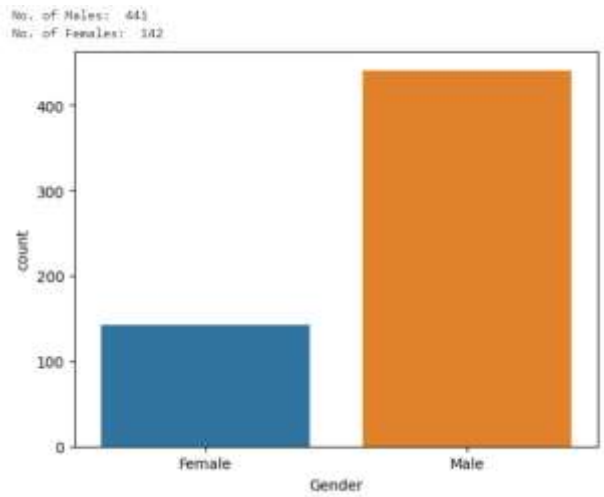
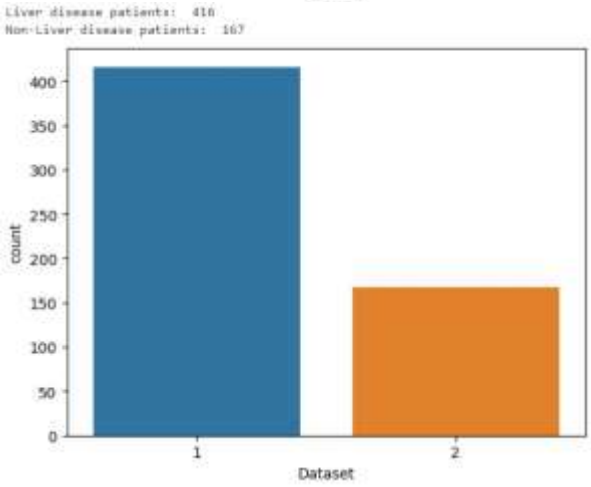
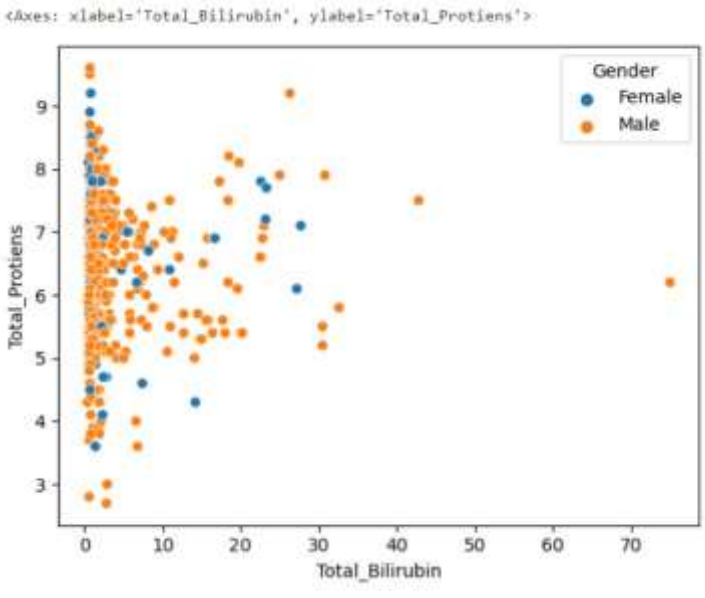
Date	12 july 2024
Team ID	739952
Project Title	Prediction and Analysis of Liver Patient Data Using Machine Learning
Maximum Marks	6 Marks

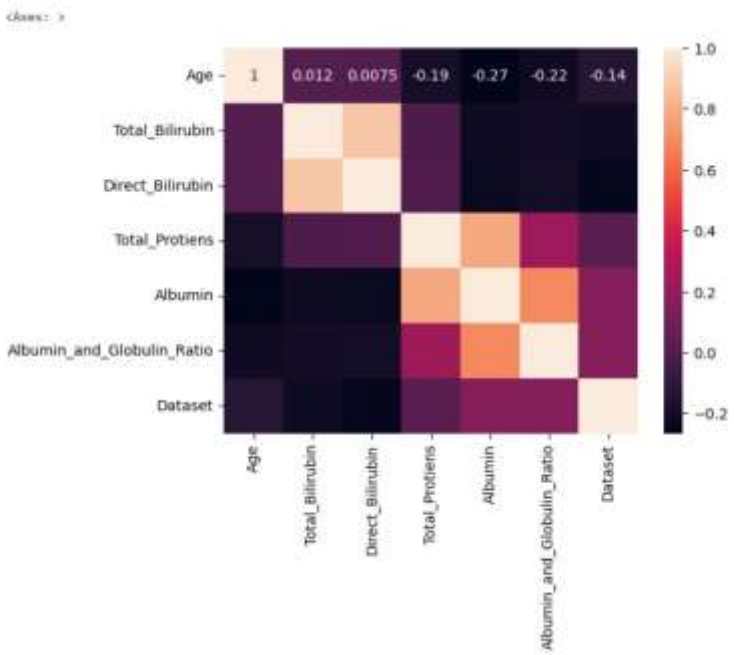
Data Exploration and Preprocessing Template

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	583 rows × 11 columns 

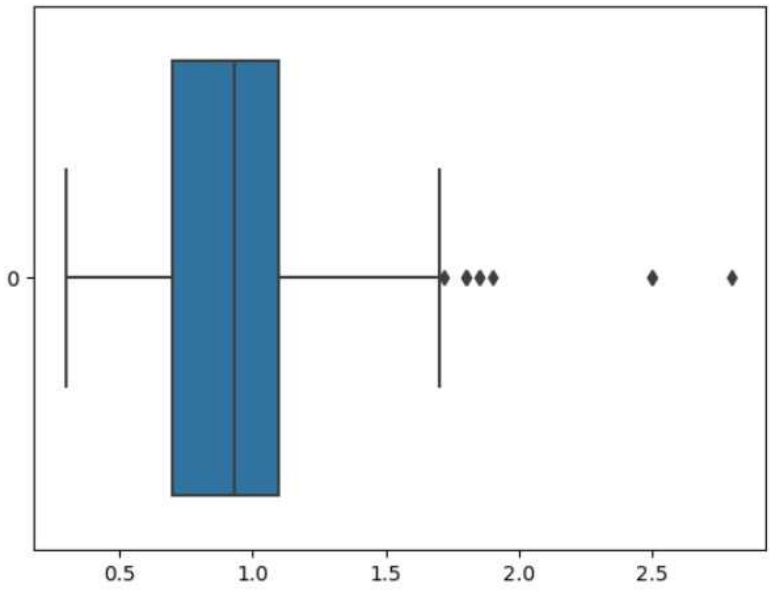
Univariate Analysis	 

Bivariate Analysis	 
Multivariate Analysis	

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>Outliers and Anomalies</td><td>

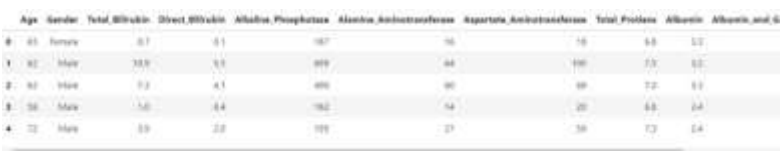
```
sns.boxplot(data.Albumin_and_Globulin_Ratio,orient='h')
```

<Axes: >


</td></tr>
<tr><td colspan="2">Data Preprocessing Code Screenshots</td></tr>
<tr><td>Loading Data</td><td>

```
# loading the dataset
data = pd.read_csv("indian_liver_patient.csv")
```

</td></tr>
</table>

<table>
<tr><td></td><td>  </td></tr>
<tr><td>Handling Missing Data</td><td>

```
data.isnull().sum()
```

```
Age                         0
Gender                      0
Total_Bilirubin             0
Direct_Bilirubin            0
Alkaline_Phosphotase        0
Alamine_Aminotransferase    0
Aspartate_Aminotransferase  0
Total_Protiens              0
Albumin                     0
Albumin_and_Globulin_Ratio  4
Dataset                     0
dtype: int64
```

```
data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].mode()[0],inplace=True)
```

```
data.isna().sum()
```

```
Age                         0
Gender                      0
Total_Bilirubin             0
Direct_Bilirubin            0
Alkaline_Phosphotase        0
Alamine_Aminotransferase    0
Aspartate_Aminotransferase  0
Total_Protiens              0
Albumin                     0
Albumin_and_Globulin_Ratio  0
Dataset                     0
dtype: int64
```

</td></tr>
<tr><td>Data Transformation</td><td>

```
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
x=sc.fit_transform(x)
```

```
x
```

```
array([[ 1.25209764, -1.76228085, -0.41887783, ...,  0.29211961,
         0.19896867, -0.14789798],
       [ 1.06663704,  0.56744644,  1.22517135, ...,  0.93756634,
         0.07315659, -0.65069686],
       [ 1.06663704,  0.56744644,  0.6449187 , ...,  0.47653296,
         0.19896867, -0.17932291],
       ...,
       [ 0.44843504,  0.56744644, -0.4027597 , ..., -0.0767071 ,
         0.07315659,  0.16635131],
       [-0.84978917,  0.56744644, -0.32216986, ...,  0.29211961,
         0.32478075,  0.16635131],
       [-0.41704777,  0.56744644, -0.37052344, ...,  0.75315299,
         1.58290153,  1.73759779]])
```

</td></tr>
<tr><td>Feature Engineering</td><td>  </td></tr>
</table>

Save Processed Data	``` import pickle pickle.dump(svm , open('model.pkl','wb')) pickle.dump(sc , open('sc.pkl','wb')) ```