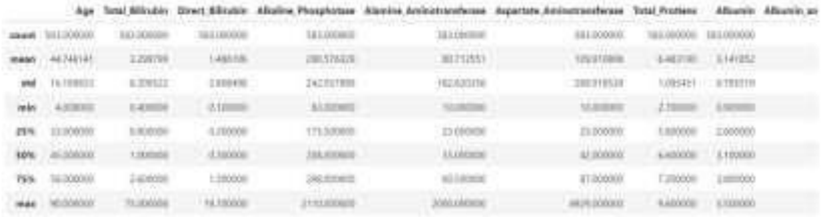
Data Collection and Preprocessing Phase

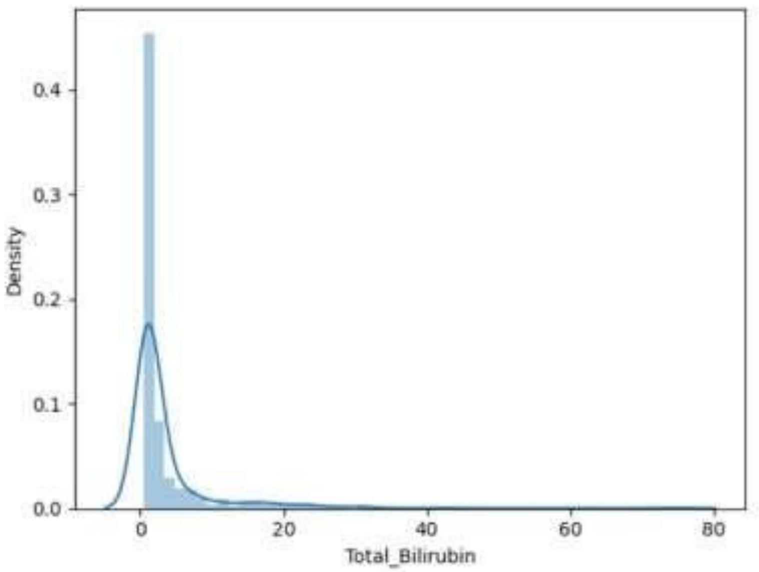
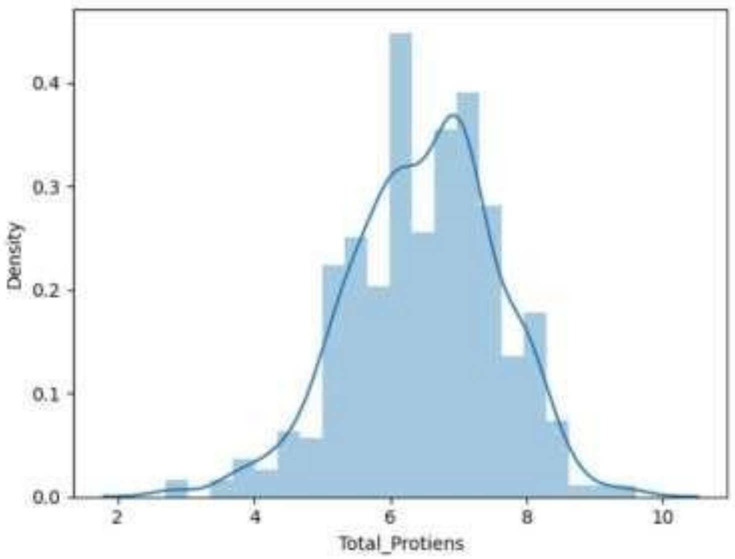
Date	12 july 2024
Team ID	739651
Project Title	Prediction and Analysis of Liver Patient Data Using Machine Learning
Maximum Marks	6 Marks

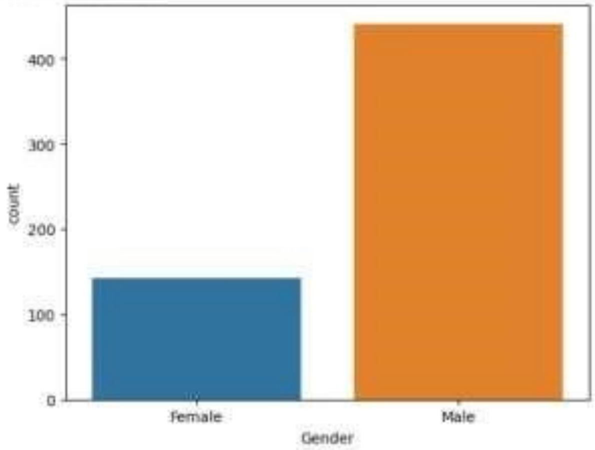
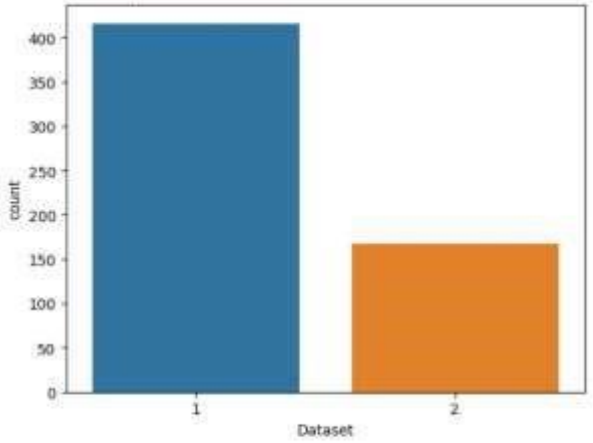
Data Exploration and Preprocessing Template

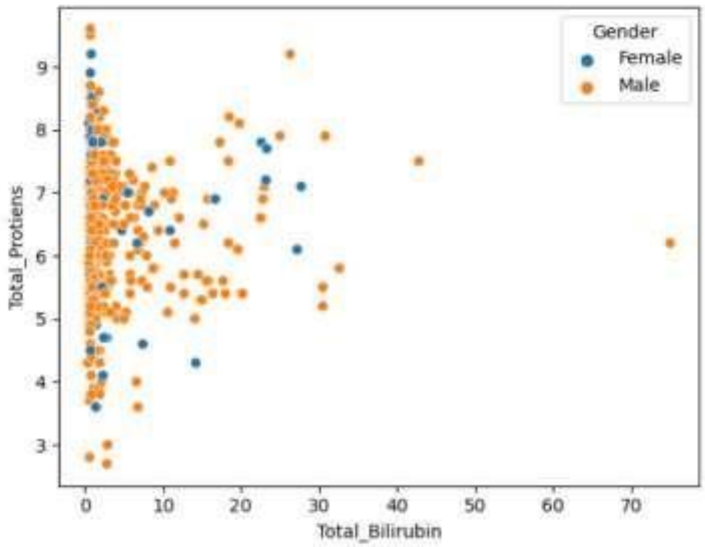
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

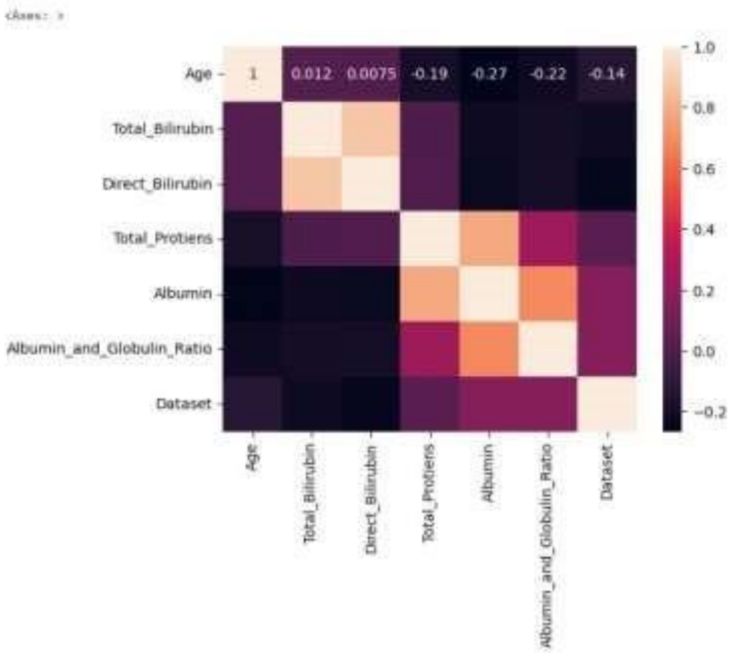
Section	**Description**
Data Overview	583 rows × 11 columns 

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_an
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.141852	
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.800000	
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	

Univariate Analysis	<Axes: xlabel='Total_Bilirubin', ylabel='Density'>  <Axes: xlabel='Total_Protiens', ylabel='Density'> 

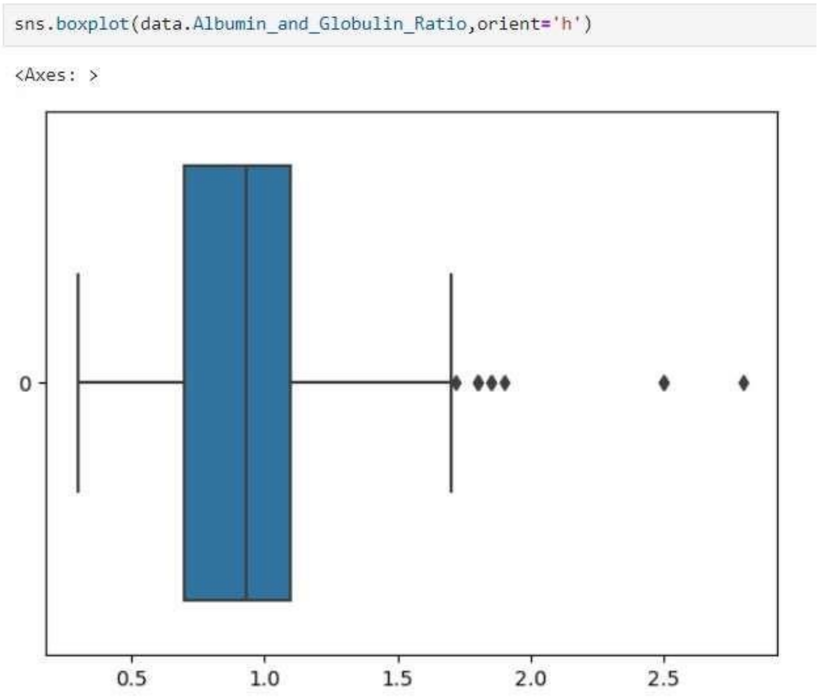
Bivariate Analysis	No. of Males: 441 No. of Females: 142  Liver disease patients: 416 Non-Liver disease patients: 167 

Multivariate Analysis	`<Axes: xlabel='Total_Bilirubin', ylabel='Total_Protiens'>` 



```
<Axes: >
```

Outliers and Anomalies



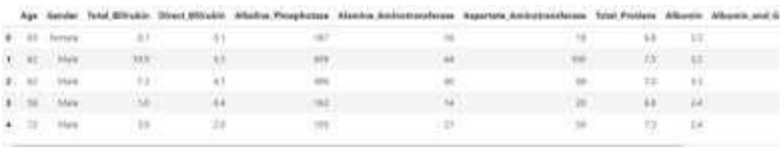
```
sns.boxplot(data.Albumin_and_Globulin_Ratio,orient='h')
```

```
<Axes: >
```

Data Preprocessing Code Screenshots

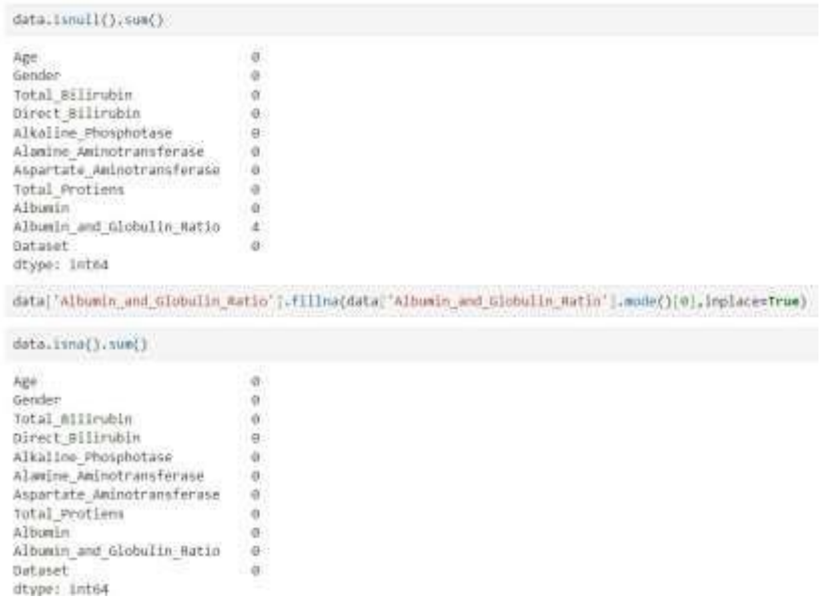
Loading Data

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



	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_G
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	

Handling Missing Data



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

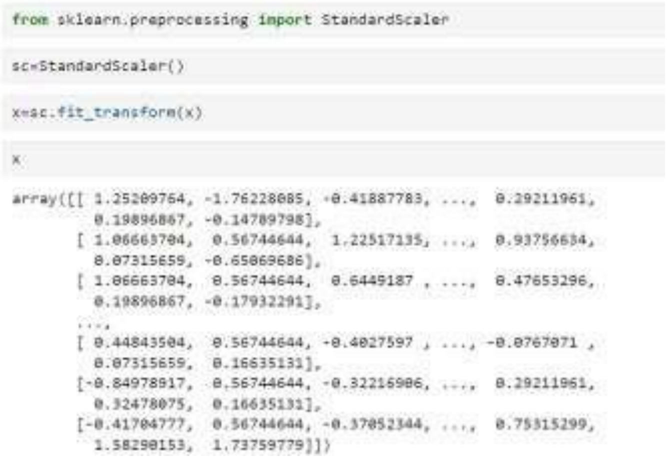
```
Age                           0
Gender                        0
Total_Bilirubin               0
Direct_Bilirubin              0
Alkaline_Phosphotase          0
Alamine_Aminotransferase      0
Aspartate_Aminotransferase    0
Total_Protiens                0
Albumin                       0
Albumin_and_Globulin_Ratio    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
```

Data Transformation



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

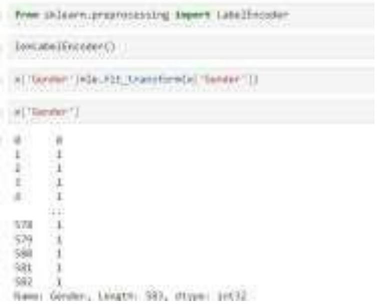
```
sc=StandardScaler()
```

```
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]])
```

Feature Engineering



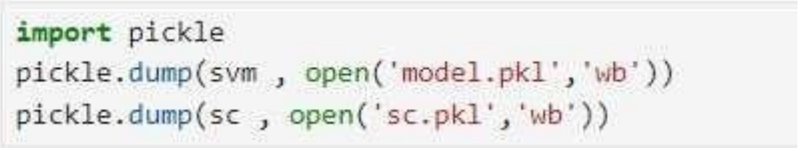
```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
x['Gender']=le.fit_transform(x['Gender'])
```

```
x['Gender']
```

```
0      0
1      1
2      1
3      1
4      1
      ..
578    1
579    1
580    1
581    1
582    1
Name: Gender, Length: 583, dtype: int32
```

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