

data.head()

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

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   Age                                   583 non-null    int64
 1   Gender                               583 non-null    object
 2   Total_Bilirubin                      583 non-null    float64
 3   Direct_Bilirubin                    583 non-null    float64
 4   Alkaline_Phosphotase                 583 non-null    int64
 5   Alanine_Aminotransferase             583 non-null    int64
 6   Aspartate_Aminotransferase           583 non-null    int64
 7   Total_Proteins                       583 non-null    float64
 8   Albumin                              583 non-null    float64
 9   Albumin_and_Globulin_Ratio          579 non-null    float64
10   Dataset                              583 non-null    int64
dtypes: float64(5), int64(5), object(1)
memory usage: 58.2+ KB
```

```
data.isnull().any()
```

Age	False
Gender	False
Total_Bilirubin	False
Direct_Bilirubin	False
Alkaline_Phosphotase	False
Alamine_Aminotransferase	False
Aspartate_Aminotransferase	False
Total_Protiens	False
Albumin	False
Albumin_and_Globulin_Ratio	True
Dataset	False
dtype:	bool

```
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'] = data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
data.isnull().sum()
```

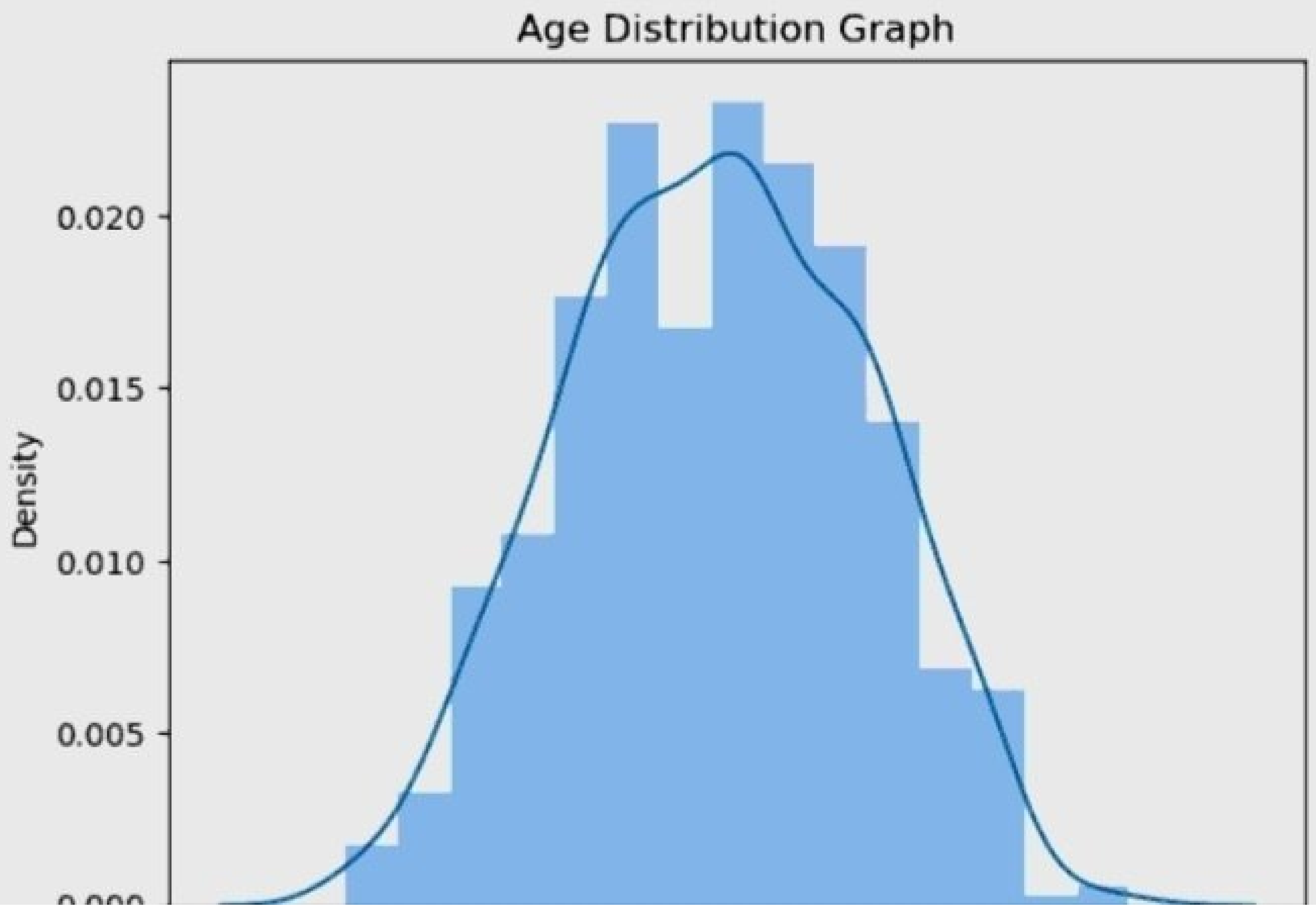
Age	0
Gender	0
Total_Bilirubin	0
Direct_Bilirubin	0
Alkaline_Phosphotase	0
Alamine_Aminotransferase	0
Aspartate_Aminotransferase	0
Total_Protiens	0
Albumin	0
Albumin_and_Globulin_Ratio	0
Dataset	0
dtype: int64	

```
2  
1 data.describe()
```

	age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alanine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000
mean	44.746141	3.296799	1.486106	290.578329	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

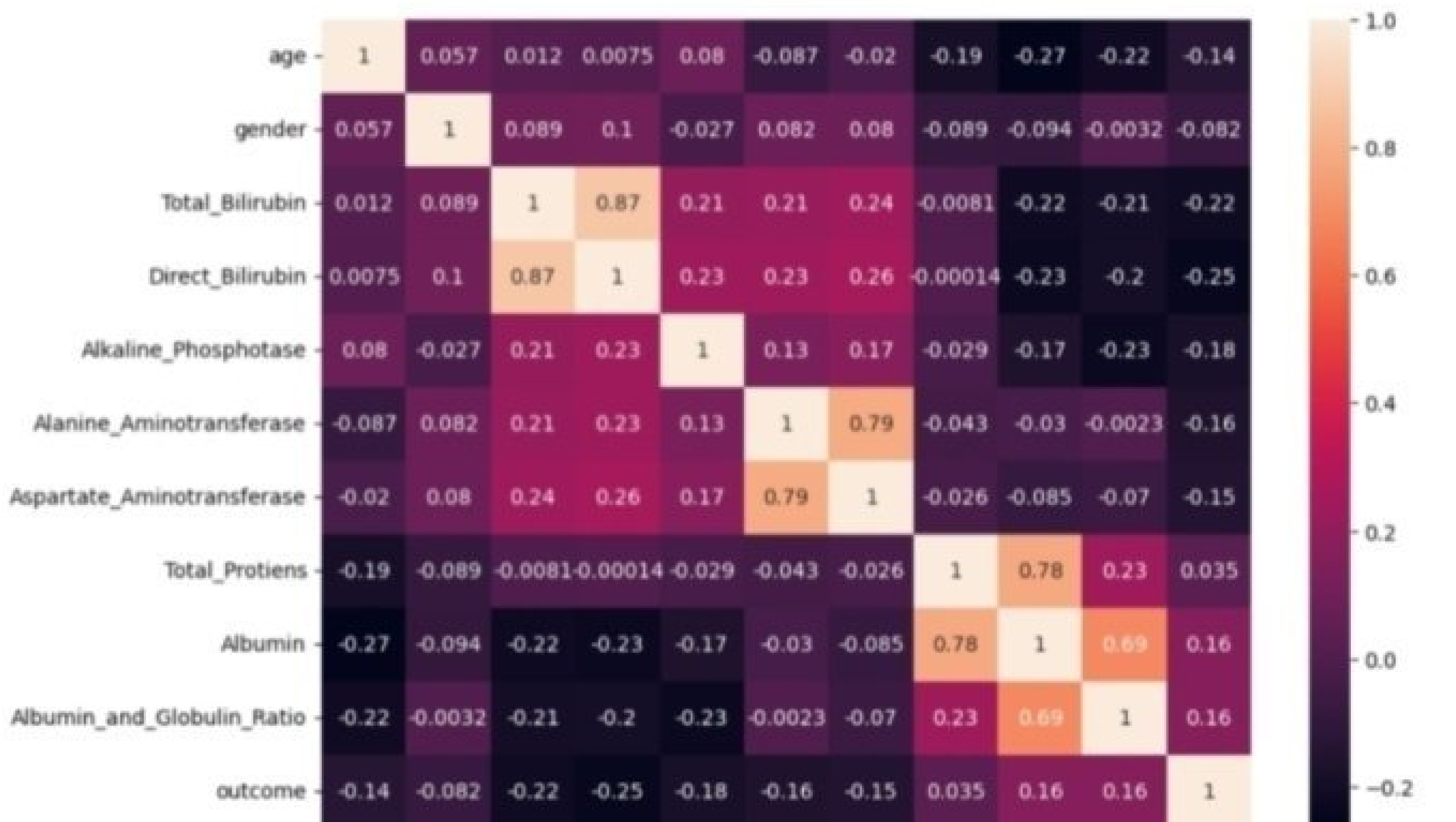
```
1 sns.distplot(data['age'])
2 plt.title('Age Distribution Graph')
3 plt.show()
4
```

D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is deprecated in a future version. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)



```
1 plt.figure(figsize=(10,7))
2
3 sns.heatmap(df.corr(),annot=True)
```

<AxesSubplot:>

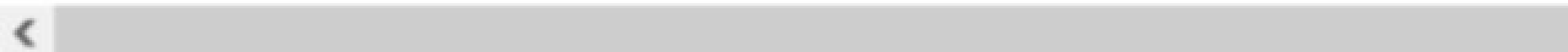




1

X\_scaled.head()

	age	gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase
0	1.252098	-1.762281	-0.418878	-0.493964	-0.426715
1	1.066637	0.567446	1.225171	1.430423	1.682629
2	1.066637	0.567446	0.644919	0.931508	0.821588
3	0.819356	0.567446	-0.370523	-0.387054	-0.447314
4	1.684839	0.567446	0.096902	0.183135	-0.393756



1	y_train.value_counts()
1	329
0	137
Name: outcome, dtype: int64	

```
1 y_train_smote.value_counts()
```

```
1    329
```

```
0    329
```

```
Name: outcome, dtype: int64
```

```
> model_history = classifier.fit(X_train, y_train, batch_size=100, validation_split=0.2, epochs=100)

Epoch 1/100
4/4 [-----] - 2s 133ms/step - loss: 0.6497 - accuracy: 0.6532 - val_loss: 0.6394 - val_accuracy: 0.7234
Epoch 2/100
4/4 [-----] - 0s 21ms/step - loss: 0.6112 - accuracy: 0.7070 - val_loss: 0.6062 - val_accuracy: 0.7234
Epoch 3/100
4/4 [-----] - 0s 20ms/step - loss: 0.5901 - accuracy: 0.7016 - val_loss: 0.5800 - val_accuracy: 0.7234
Epoch 4/100
4/4 [-----] - 0s 20ms/step - loss: 0.5743 - accuracy: 0.7016 - val_loss: 0.5592 - val_accuracy: 0.7234
Epoch 5/100
4/4 [-----] - 0s 21ms/step - loss: 0.5619 - accuracy: 0.7016 - val_loss: 0.5437 - val_accuracy: 0.7234
```

```

1 #Age--Gender--Total_Bilirubin--Direct_Bilirubin--Alkaline_Phosphotase--Alanin_Aminotransferase --Asparate_Aminotrans
2 model4.predict([[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]) : 0.77
< >
D:\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClass
fier was fitted with feature names : 0.75
warnings.warn(
array([1], dtype=int64) : 0.76
1 #Age--Gender--Total_Bilirubin--Direct_Bilirubin--Alkaline_Phosphotase--Alanin_Aminotransferase --Asparate_Aminotrans
2 model1.predict([[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]) : 0.77
< >
D:\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomForestClass : 0.77
fier was fitted with feature names
warnings.warn(

```

```
1 y_pred = classifier.predict(X_test)
```

```
4/4 [=====] - @s 2ms/step
```

---

```
1 y_pred
```

```
y_pred = (y_pred > 0.5)  
y_pred
```

```
y([[ True],  
   [ True],  
   [ True],  
   [ True]).
```

```
1 #Age→Gender→Total_Bilirubin→Direct_Bilirubin→Alkaline_Phosphotase→
2 sample_value = [[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
3 if predict_exit(sample_value)>0.5:
4     print('Prediction: Liver Patient')
5 else:
6     print('Prediction: Healthy ')
<
```

1/1 [=====] - 0s 105ms/step

Prediction: Liver Patient



```
1 acc_smote= [['KNN Classifier', knn1], ['RandomForestClassifier', rfc1],
2            ['DecisionTreeClassifier', dtc1], ['LogisticRegression', logi1]]
3 Liverpatient_pred= pd.DataFrame(acc_smote, columns = ['classification models', 'accuracy_score'])
4 Liverpatient_pred
```

	classification models	accuracy_score
0	KNN Classifier	0.555556
1	RandomForestClassifier	0.709402
2	DecisionTreeClassifier	0.683761
3	LogisticRegression	0.641026

```
1 from sklearn.ensemble import ExtraTreesClassifier
2 model = ExtraTreesClassifier()
3 model.fit(X,y)
```

ExtraTreesClassifier()

---

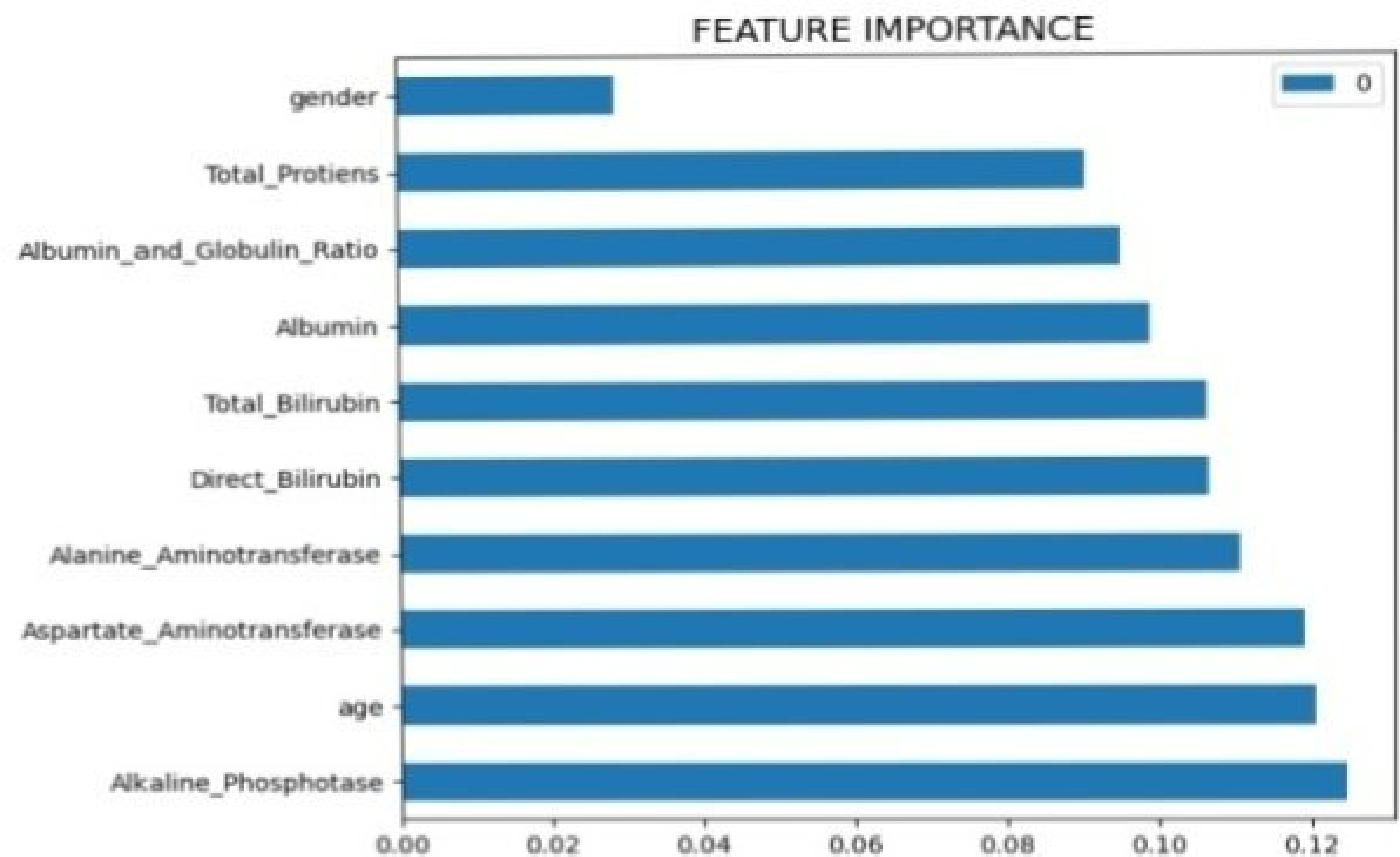
```
1 model.feature_importances_
array([0.1205029 , 0.02863187, 0.10625368, 0.10648548, 0.1245292 ,
       0.11049943, 0.118963  , 0.09033392, 0.09882431, 0.09497621])

1 dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascending=False)
2 dd
```

	0
Alkaline_Phosphotase	0.124529
age	0.120503
Aspartate_Aminotransferase	0.118963
Alanine_Aminotransferase	0.110499
Direct_Bilirubin	0.106485
Total_Bilirubin	0.106254
Albumin	0.098824
Albumin_and_Globulin_Ratio	0.094976
Total_Protiens	0.090334
gender	0.028632

```
1 dd.plot(kind='barh', figsize=(7,6))
2 plt.title("FEATURE IMPORTANCE",fontsize=14)
```

```
text(0.5, 1.0, 'FEATURE IMPORTANCE')
```



```
1 import joblib
2 joblib.dump(model1, 'ETC.pkl')
```

```
['ETC.pkl']
```

---