SKILL ORIENTED COURSE-2

STATISTICAL ANALYSIS AND DATA ANALYTICS APPLICATIONS ON LIVER DATASET

A Skill Oriented Seminar Report Submitted in Partial Fulfilment of The Requirements for an award of

INFORMATION TECHNOLOGY

By

Conducted by APSSDC- CM's Centre of Excellence

&

Organised by Department of **IT**

ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An Autonomous Institute)

(Approved by AICTE, Permanently Affiliated to JNTU Kakinada, Accredited by NBA & NAAC A+

Recognized by UGC under Section 2(f) & 12(B)) **TEKKALI , ANDHRA PRADESH.**

ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Skill Oriented Course entitled “Data Analysis Through Python” is being submitted by JAMI SAI KAMAL (21A51A1231),K GUNA (21A51A1236), SASANAPURI TARUN (21A51A1256), Y JASWANTH (21A51A1264) in partial fulfilment of requirements for the award of "DATA ANALYSIS" project in INFORMATION TECHNOLOGY, ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Tekkali is a record of work carried out by the team members during the academic year 2021-2022.

Signature of the Co-ordinator	**Signature of Head of the Department**
Dr. T. PANDU RANGA VITAL	Dr.YEGIREDDI RAMESH,
Asso.professor	Head of the Department
Department of IT	Department of IT
M.Tech,Ph.D	M.Tech.,Ph.D..

TABLE OF CONTENTS

ABSTRACT

The liver dataset is a popular dataset in the field of machine learning and data analysis.It contains total bilirubin, direct bilirubin, total proteins, and albumin are important markers that provide information about liver function and overall liver health.This data analysis project aims to explore the liver dataset in detail, and to identify patterns and relationships between the variables

The first step in the analysis is to perform exploratory data analysis (EDA) to gain a better understanding of the dataset. EDA is a technique to analyze data using some visual techniques.With this technique, we can get detailed information about he statistical summary of data.The EDA includes visualizations such as scatter plots, histograms, and box plots to examine the distribution and relationships between the variables.The scatter plot matrix is particularly useful in identifying patterns and correlations between the variables. From the EDA, we can see that there is a strong positive correlation between total_bilirubin and direct_bilirubin weaker positive correlation between total_protein and age and we will also be able to deal with the duplicate values, outliers, and also see some trends or patterns present in the “liver dataset”

Data collection is the first step in data analysis, and it involves gathering data from various sources such as surveys,experiments,and databases.The quality of the data collected is critical,as it can affect the accuracy and reliability of the analysis.Data cleaning is the process of removing errors, inconsistencies, and missing values from the data. This step is essential for ensuring that the data is accurate and complete.

This data analysis project explores the liver dataset, which contains functions like tot_bilirubin,direct_bilirubin,tot_proteins,albumin,sgot,sgpt and alkphos to describe liver condition. The study aims to identify patterns and relationships between the variables,and to classify the liver based on their functions. The results of the analysis show that the tot_bilirubin and direct_bilirubin are the most important features for classifying the liver disease.The findings of this study can be used to improve our understanding of the liver condition and to develop more accurate classification models for other datasets.[2]

INTRODUCTION TO DATA ANALYSIS

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively. Data mining is a particular data analysis technique that focuses on statistical modelling and knowledge discovery for predictive rather than purely descriptive purposes, while business intelligence covers data analysis that relies heavily on aggregation, focusing mainly on business information. In statistical applications, data analysis can be divided into descriptive statistics, exploratory data analysis (EDA), and confirmatory data analysis (CDA). EDA focuses on discovering new features in the data while CDA focuses on confirming or falsifying existing hypotheses. Predictive analytics focuses on the application of statistical models for predictive forecasting or classification, while text analytics applies statistical, linguistic, and structural techniques to extract and classify information from textual sources, a species of unstructured data. All of the above are varieties of data analysis. Data integration is a precursor to data analysis, and data analysis is closely linked to data visualization and data dissemination.

The goal of data analysis is to transform raw data into useful information that can be used to make informed decisions. This involves several steps, including data collection, data cleaning, data transformation, data modeling, and data visualization. Each of these steps is essential for ensuring that the data is accurate, complete, and relevant to the problem at hand.

Data collection is the first step in data analysis, and it involves gathering data from various sources such as surveys, experiments, and databases. The quality of the data collected is critical, as it can affect the accuracy and reliability of the analysis. Data cleaning is the process of removing errors, inconsistencies, and missing values from the data. This step is essential for ensuring that the data is accurate and complete.

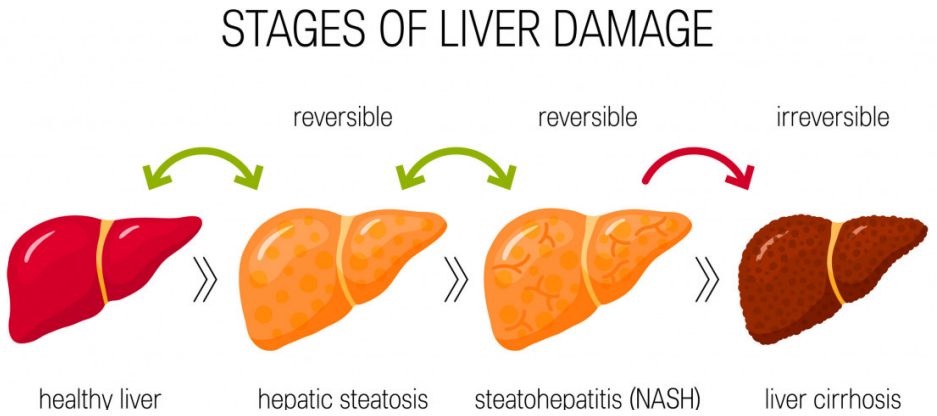
Data transformation involves converting the data into a format that is suitable for analysis. This may involve aggregating data, creating new variables, or normalizing data.[3]

ABOUT LIVER DATASET

The liver Dataset contains four main functions total_bilirubin, direct_bilirubin, total_proteins, albumin and alkphos. These functions were used to create a linear discriminant model to classify the gender.The dataset is often used in data mining, classification and clustering examples and to test algorithms.

The liver is an essential organ located in the upper right side of the abdomen. It plays a vital role in various metabolic processes, including detoxification, protein synthesis, bile production, and storage of vitamins and minerals. Liver disease refers to any condition that affects the structure or function of the liver, impairing its ability to perform these critical functions.

Just for reference, here are pictures of liver diseases:

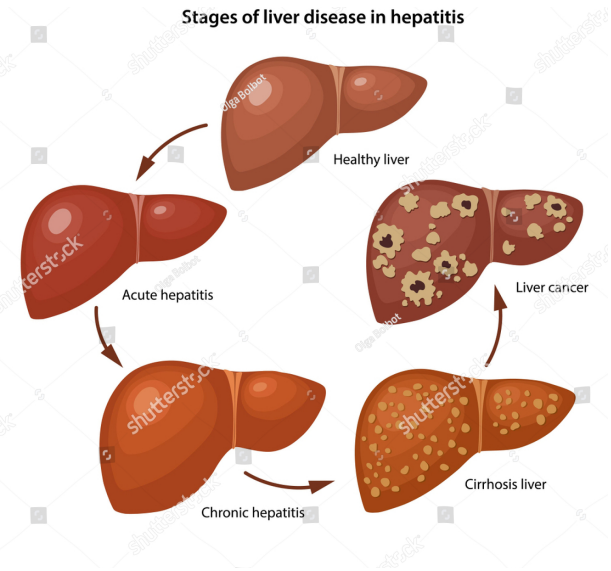


Liver diseases can have numerous causes, including viral infections (such as hepatitis viruses), excessive alcohol consumption, autoimmune disorders, genetic disorders, metabolic disorders, drug-induced liver injury, fatty liver disease, and cirrhosis. They can range from mild, temporary conditions to severe and chronic diseases that can significantly impact a person's health and quality of life.

There are several types of liver diseases, each with its own characteristics and consequences. Some common liver diseases include:

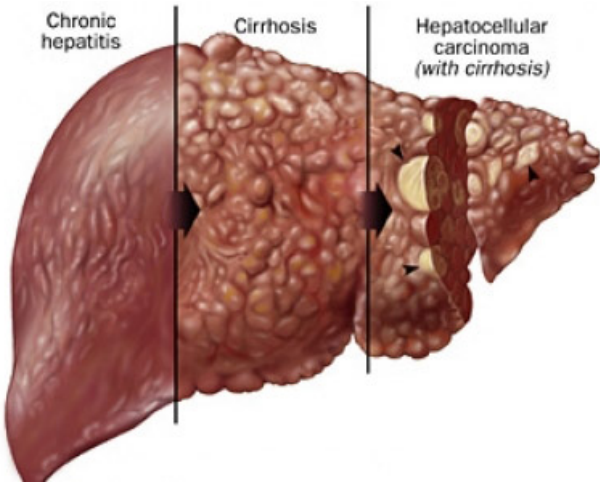
1. Hepatitis: Hepatitis is inflammation of the liver, usually caused by viral infections (hepatitis A, B, C, D, or E). It can also be caused by autoimmune conditions, alcohol abuse, certain medications, or toxins.

here's a picture of stages of liver disease in hepatits:



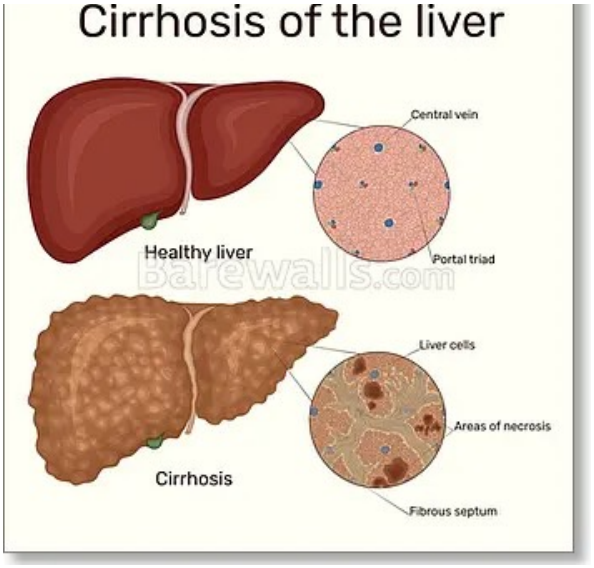
Cirrhosis: Cirrhosis is the advanced stage of liver disease characterized by extensive scarring (fibrosis) of the liver tissue. It is commonly caused by chronic liver diseases such as hepatitis, excessive alcohol consumption, or non-alcoholic fatty liver disease. Cirrhosis can lead to liver failure if not properly managed.

here's a picture of cirrhosis of liver:



Liver cancer: Liver cancer, or hepatocellular carcinoma, is a malignant tumor that originates in the liver. It can occur as a primary cancer or as a result of metastasis from other organs.

here's a picture of liver cancer:[5]



In the context of liver disease, total bilirubin, direct bilirubin, total proteins, and albumin are important markers that provide information about liver function and overall liver health.

Bilirubin is a yellow pigment that is produced as a byproduct of the breakdown of red blood cells. It is metabolized by the liver and excreted in bile. In liver disease, the liver may be unable to effectively process bilirubin, leading to an accumulation in the bloodstream. This elevation in bilirubin levels can result in jaundice, a yellowing of the skin and eyes.

Total bilirubin refers to the sum of both direct bilirubin and indirect bilirubin. Indirect bilirubin is the unconjugated form that circulates in the bloodstream, while direct bilirubin is the conjugated form that has been processed by the liver.

Direct bilirubin levels are particularly useful in assessing liver function as they indicate how well the liver is able to conjugate bilirubin and excrete it in bile. Elevated levels of direct bilirubin can suggest liver diseases such as hepatitis, cholestasis (impaired bile flow), or other conditions affecting the liver.

Total proteins and albumin are markers that reflect the liver's ability to synthesize proteins. Albumin is the most abundant protein synthesized by the liver and plays a crucial role in maintaining the osmotic pressure of blood and transporting various substances. In liver disease, there may be a decrease in albumin production, resulting in low levels of albumin in the blood. Total protein levels can also be affected, as they include albumin as well as other proteins synthesized by the liver.

Low levels of total proteins and albumin can indicate liver dysfunction and impaired liver synthetic function. Additionally, other factors such as malnutrition or kidney disease can contribute to low levels of these proteins.

It's important to note that these markers are just a few of the many tests used to evaluate liver function. Other tests, such as liver enzymes (AST, ALT, ALP), coagulation factors, and imaging studies, are typically used in conjunction with these markers to provide a comprehensive assessment of liver health and aid in the diagnosis of liver disease.[6]

Load the Dataset

In [1]:
```
# import the data processing and visualization libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
# read the dataset in pandas
df_liver=pd.read_csv("liver.csv")
```

Quick summary of Dataset

In [3]:
```
# Access the first five rows from dataset
df_liver.head()
```

Out[3]:

	age	gender	tot_bilirubin	direct_bilirubin	tot_proteins	albumin	ag_ratio	sgpt	sgot	alkphos	is_patient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1

In [4]:
```
# Access the last five rows from the dataset
df_liver.tail()
```

Out[4]:

	age	gender	tot_bilirubin	direct_bilirubin	tot_proteins	albumin	ag_ratio	sgpt	sgot	alkphos	is_patient
578	60	Male	0.5	0.1	500	20	34	5.9	1.6	0.37	2
579	40	Male	0.6	0.1	98	35	31	6.0	3.2	1.10	1
580	52	Male	0.8	0.2	245	48	49	6.4	3.2	1.00	1
581	31	Male	1.3	0.5	184	29	32	6.8	3.4	1.00	1
582	38	Male	1.0	0.3	216	21	24	7.3	4.4	1.50	2

In [5]:
```
# retrieve the column information
df_liver.columns.values
```

Out[5]:
```
array(['age', 'gender', 'tot_bilirubin', 'direct_bilirubin',
       'tot_proteins', 'albumin', 'ag_ratio', 'sgpt', 'sgot', 'alkphos',
       'is_patient'], dtype=object)
```

In [6]:
```
# Retrieve the full information of dataset regarding the features and response, in order
# if the values are unique or are there any missing data.
df_liver.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   tot_bilirubin     583 non-null    float64
 3   direct_bilirubin  583 non-null    float64
 4   tot_proteins      583 non-null    int64  
 5   albumin           583 non-null    int64  
 6   ag_ratio          583 non-null    int64  
 7   sgpt              583 non-null    float64
 8   sgot              583 non-null    float64
 9   alkphos           579 non-null    float64
 10  is_patient        583 non-null    int64  
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
```

Dimension of the dataset

In [7]:
```
#finding the shape of the dataframe
print(df_liver.shape)
```

```
(583, 11)
```

What we can see here is that the data contains (583 rows by 11 columns). This means the that df_liver contains 583 observations + 10 features + 1 response (or target) variable. The response variable is "Dataset". Furthermore, df_liver consists of 5 Floats, 5 integers and 1 object. Therefore, the goal is to convert the object to numerical values so we can apply machine learning (ML) algorithms. We also notice that the column, 'Albumin_and_Globulin_Ratio' contains missing values (Nan).

In [8]:
```
# Statistical summary using .describe()
```

In [9]:
```
print(df_liver.describe())
```

```
              age  tot_bilirubin  direct_bilirubin  tot_proteins      albumin  \
count  583.000000     583.000000        583.000000    583.000000   583.000000   
mean    44.746141       3.298799          1.486106    290.576329    80.713551   
std     16.189833       6.209522          2.808498    242.937989   182.620356   
min      4.000000       0.400000          0.100000     63.000000    10.000000   
25%     33.000000       0.800000          0.200000    175.500000    23.000000   
50%     45.000000       1.000000          0.300000    208.000000    35.000000   
75%     58.000000       2.600000          1.300000    298.000000    60.500000   
max     90.000000      75.000000         19.700000   2110.000000  2000.000000   

          ag_ratio         sgpt         sgot      alkphos  is_patient  
count   583.000000  583.000000  583.000000  579.000000  583.000000  
mean    109.910806    6.483190    3.141852    0.947064    1.286449  
std     288.918529    1.085451    0.795519    0.319592    0.452490  
min      10.000000    2.700000    0.900000    0.300000    1.000000  
25%      25.000000    5.800000    2.600000    0.700000    1.000000  
50%      42.000000    6.600000    3.100000    0.930000    1.000000  
75%      87.000000    7.200000    3.800000    1.100000    2.000000  
max    4929.000000    9.600000    5.500000    2.800000    2.000000  
```

From the descriptive statistics above, we notice that the minimum age is 4 and the maximum is 90. Based on the information on this dataset, it was suggested that anyone above the age of 85 should be treated as 90. So we can change that through the creation of a new dataframe. Furthermore, we notice missing values

in the column "Albumin_and_Globulin_Ratio", which we can deal with shortly. Lastly, it would be a good idea to figure out the ranges of healthy patients in order to figure out where each patient lies.

Satistical opeartions on data

In [10]:
```
df_liver.sum()
```

Out[10]:
```
age                                                            26087
gender              FemaleMaleMaleMaleMaleMaleFemaleFemaleMaleMale...
tot_bilirubin                                                1923.2
direct_bilirubin                                              866.4
tot_proteins                                                 169406
albumin                                                       47056
ag_ratio                                                      64078
sgpt                                                         3779.7
sgot                                                         1831.7
alkphos                                                      548.35
is_patient                                                      750
dtype: object
```

In [11]:
```
df_liver.mean()
```

```
/var/folders/m7/pk43vrzs6lg0wx91hgcx8bxw0000gn/T/ipykernel_3086/2079664606.py:1: FutureW
arning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only valid columns
before calling the reduction.
  df_liver.mean()
```

Out[11]:
```
age                  44.746141
tot_bilirubin         3.298799
direct_bilirubin      1.486106
tot_proteins        290.576329
albumin              80.713551
ag_ratio            109.910806
sgpt                  6.483190
sgot                  3.141852
alkphos               0.947064
is_patient            1.286449
dtype: float64
```

In [12]:
```
df_liver.median()
```

```
/var/folders/m7/pk43vrzs6lg0wx91hgcx8bxw0000gn/T/ipykernel_3086/2903936268.py:1: FutureW
arning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only valid columns
before calling the reduction.
  df_liver.median()
```

Out[12]:
```
age                  45.00
tot_bilirubin         1.00
direct_bilirubin      0.30
tot_proteins        208.00
albumin              35.00
ag_ratio             42.00
sgpt                  6.60
sgot                  3.10
alkphos               0.93
is_patient            1.00
dtype: float64
```

In [13]:
```
df_liver.mode()
```

Out[13]:

	age	gender	tot_bilirubin	direct_bilirubin	tot_proteins	albumin	ag_ratio	sgpt	sgot	alkphos	is_patient
0	60.0	Male	0.8	0.2	198	25.0	23.0	7.0	3.0	1.0	1.0
1	NaN	NaN	NaN	NaN	215	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	298	NaN	NaN	NaN	NaN	NaN	NaN

In [14]:
```
df_liver.std()
```

```
/var/folders/m7/pk43vrzs6lg0wx91hgcx8bxw0000gn/T/ipykernel_3086/1710450797.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df_liver.std()
```

Out[14]:
```
age                 16.189833
tot_bilirubin        6.209522
direct_bilirubin     2.808498
tot_proteins       242.937989
albumin            182.620356
ag_ratio           288.918529
sgpt                 1.085451
sgot                 0.795519
alkphos              0.319592
is_patient           0.452490
dtype: float64
```

In [15]:
```
df_liver.var()
```

```
/var/folders/m7/pk43vrzs6lg0wx91hgcx8bxw0000gn/T/ipykernel_3086/942502163.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df_liver.var()
```

Out[15]:
```
age                  262.110702
tot_bilirubin         38.558160
direct_bilirubin       7.887659
tot_proteins       59018.866587
albumin            33350.194438
ag_ratio           83473.916429
sgpt                   1.178205
sgot                   0.632850
alkphos                0.102139
is_patient             0.204747
dtype: float64
```

Selected column statistical operations

In [16]:
```
df_liver['age'].sum()
```

Out[16]:
```
26087
```

In [17]:
```
df_liver['tot_bilirubin'].mean()
```

Out[17]:
```
3.298799313893652
```

In [18]:
```
df_liver['tot_proteins'].median()
```

Out[18]:
```
208.0
```

In [19]:
```
df_liver['albumin'].mode()
```

Out[19]:
```
0    25
Name: albumin, dtype: int64
```

In [20]:
```
df_liver['direct_bilirubin'].std()
```

Out[20]:
```
2.8084976176589636
```

Cleaning the data

a) Healthy ranges of the feature results

Healthy Ranges for the 10 feature columns

Total_Bilirubin = 0.1 to 1.2 mg/dL = 1.71 to 20.5 umol/L

Direct_Bilirubin = < 0.3 mg/dL = < 5.1 umol/L

Alkaline_Phosphatase = 44 to 147 IU/L (High levels of ALP are seen in children undergoing growth and pregnant women)

Alamine_Aminotransferase = 29 to 33 IU/L (Age and gender can affect the value)

Aspartate_Aminotransferase = 1 to 45 U/L (Values are slightly lower in females) Total_Proteins = 6.0 to 8.3 g/dL

Albumin = 3.4 to 5.4 g/dL

Albumin_and_Globulin_Ratio = Adult: 3.7 to 5.2 g/dL; Older Adult: 3.2 to 4.6 g/dL; >90 yr: 2.9 to 4.5 g/dL

Note: These values may differ based on the different guidelines or hospitals. The values above were obtained from google.

b) Dealing with missing values

In [21]:
```
df_liver.describe(include='all')
```

Out[21]:

	age	gender	tot_bilirubin	direct_bilirubin	tot_proteins	albumin	ag_ratio	sgpt	5
count	583.000000	583	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	
unique	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	Male	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	441	NaN	NaN	NaN	NaN	NaN	NaN	
mean	44.746141	NaN	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	
std	16.189833	NaN	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	
min	4.000000	NaN	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	
25%	33.000000	NaN	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	
50%	45.000000	NaN	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	
75%	58.000000	NaN	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	
max	90.000000	NaN	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	

In [22]:
```
# Define a function that allows us to create a table of missing values in df_liver and t
# descending order
def missing_values(data):
    total = data.isnull().sum().sort_values(ascending=False)
    percentage = (data.isnull().sum()/data.isnull().count()).sort_values(ascending=False
    percentage_final = (round(percentage, 2) * 100)
    total_percent = pd.concat(objs=[total, percentage_final], axis = 1, keys=['Total', '
    return total_percent
```

In [23]:
```
# Find the total count and % of missing values
missing_values(df_liver)
```

Out[23]:

	Total	%
alkphos	4	1.0
age	0	0.0
gender	0	0.0
tot_bilirubin	0	0.0
direct_bilirubin	0	0.0
tot_proteins	0	0.0
albumin	0	0.0
ag_ratio	0	0.0
sgpt	0	0.0
sgot	0	0.0
is_patient	0	0.0

It appears that there are only 4 missing values in the feature column alkphos, which equates to 1% of the the entire data.

In [24]:
```
# Replace missing values with the mean of feature column alkphos,
# then check to see that it has been successfull, where the sum of missig values should
df_liver['alkphos'].fillna(df_liver['alkphos'].mean(), inplace = True)
df_liver['alkphos'].isnull().sum()
```

Out[24]:
```
0
```

In [25]:
```
# Repeat to see what is the % of missing values
missing_values(df_liver)
```

Out[25]:

	Total	%
age	0	0.0
gender	0	0.0
tot_bilirubin	0	0.0
direct_bilirubin	0	0.0
tot_proteins	0	0.0
albumin	0	0.0
ag_ratio	0	0.0
sgpt	0	0.0
sgot	0	0.0
alkphos	0	0.0
is_patient	0	0.0

VISUALIZATION

Data visualization is the representation of data through use of common graphics,such as charts,plots,infographics and animations

Types of Data Visualization

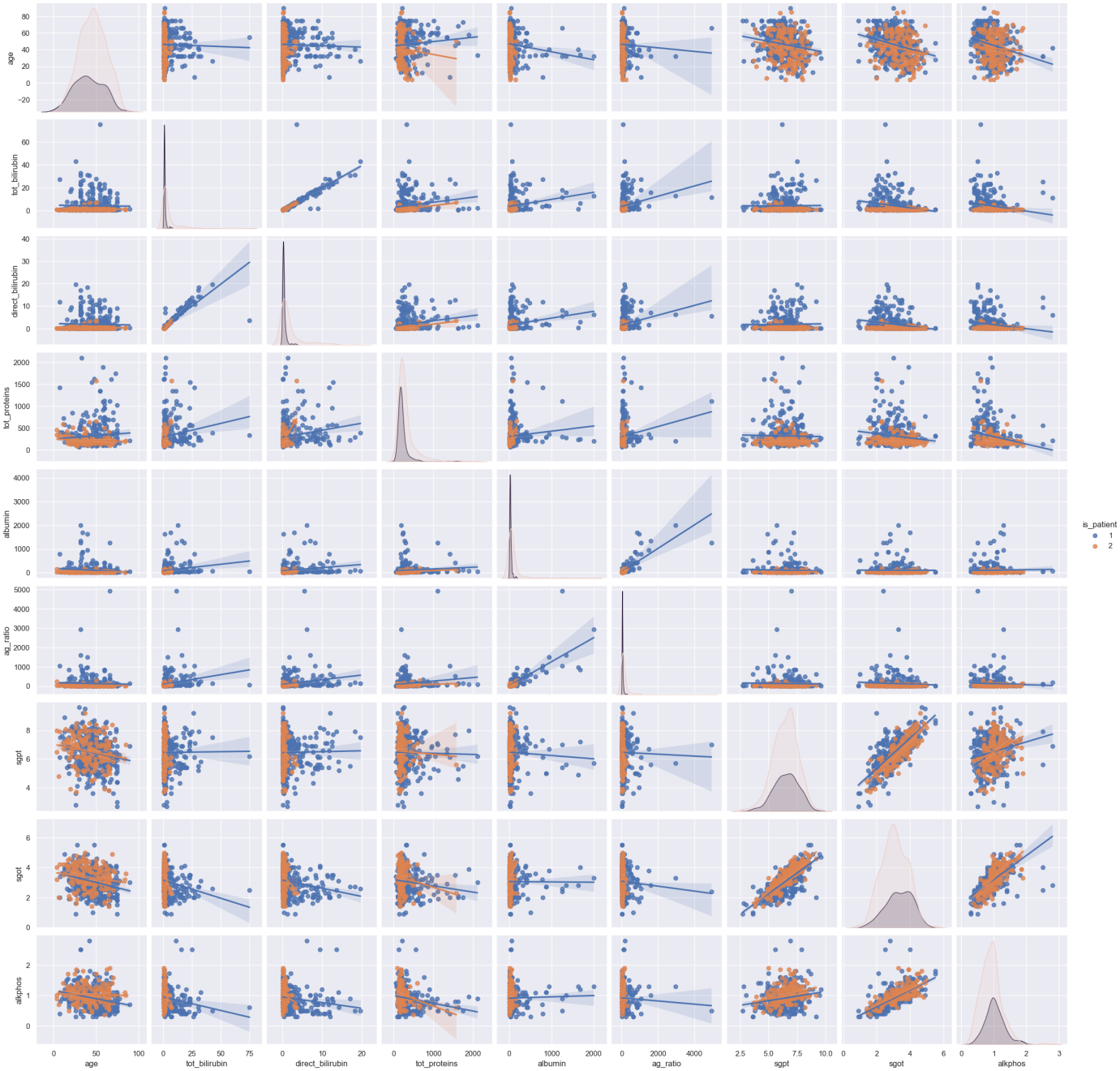
1. Tables
2. Pie Charts
3. Line Charts and Area Charts
4. Histograms
5. Scatter Plots
6. Heat Maps
7. Tree Maps

Exploring the Data Visually

Finding any corelation between the features using pairplot in seaborn

In [26]:
```
# Corelation Pairplot
sns.set()
sns.pairplot(df_liver,hue='is_patient',kind='reg')
```

Out[26]:
```
<seaborn.axisgrid.PairGrid at 0x7fabd054d0a0>
```



Result Analysis:

Based on the correlative pair plots, we find some interesting results directly.

-Positive correlations:

Total Bilirubin and Direct Bilirubin (vice-versa)

Alamine Aminotransferase and Aspartate Aminotransferase (vice-versa)

Total Protein and Albumin (vice-versa)

Albumin and Globulin Ratio and Albumin (vice-versa)

Total Protein and Albumin and Globulin Ration (vice-versa)

-Negative correlations:

Total Protein and age (vice-versa)

Albumin and age (vice-versa)

Albumin and Globulin Ration and age (vice-versa)

In [27]:
```
# A more robust way of figuring out correlations other than observations as above is to
# table with the ranging from -1 to 1
df_liver.corr().style.background_gradient(cmap='coolwarm')
```

Out[27]:

	age	tot_bilirubin	direct_bilirubin	tot_proteins	albumin	ag_ratio	sgpt	sgot
age	1.000000	0.011763	0.007529	0.080425	-0.086883	-0.019910	-0.187461	-0.265924
tot_bilirubin	0.011763	1.000000	0.874618	0.206669	0.214065	0.237831	-0.008099	-0.222250
direct_bilirubin	0.007529	0.874618	1.000000	0.234939	0.233894	0.257544	-0.000139	-0.228531
tot_proteins	0.080425	0.206669	0.234939	1.000000	0.125680	0.167196	-0.028514	-0.165453
albumin	-0.086883	0.214065	0.233894	0.125680	1.000000	0.791966	-0.042518	-0.029742
ag_ratio	-0.019910	0.237831	0.257544	0.167196	0.791966	1.000000	-0.025645	-0.085290
sgpt	-0.187461	-0.008099	-0.000139	-0.028514	-0.042518	-0.025645	1.000000	0.784053
sgot	-0.265924	-0.222250	-0.228531	-0.165453	-0.029742	-0.085290	0.784053	1.000000
alkphos	-0.216089	-0.206159	-0.200004	-0.233960	-0.002374	-0.070024	0.233904	0.686322
is_patient	-0.137351	-0.220208	-0.246046	-0.184866	-0.163416	-0.151934	0.035008	0.161388

Result Analysis:

The above correlation heatmap demonstrates strong positive (closer to 1) and negative correlations (closer to -1) but also weak positive and negative correlations (closer to zero). Next, let us plot some of these features as a function of gender in order to determine whether gender effects the target feature and the concentration levels of some of those feautures, which are deterministic of liver disease. However, before doing so we need to change the gender to numerical values.

In [29]:
```
gender_data = df_liver[['gender', 'is_patient']].groupby('gender', as_index = False).agg
gender_data
```

Out[29]:

	gender	is_patient
0	Female	192
1	Male	558

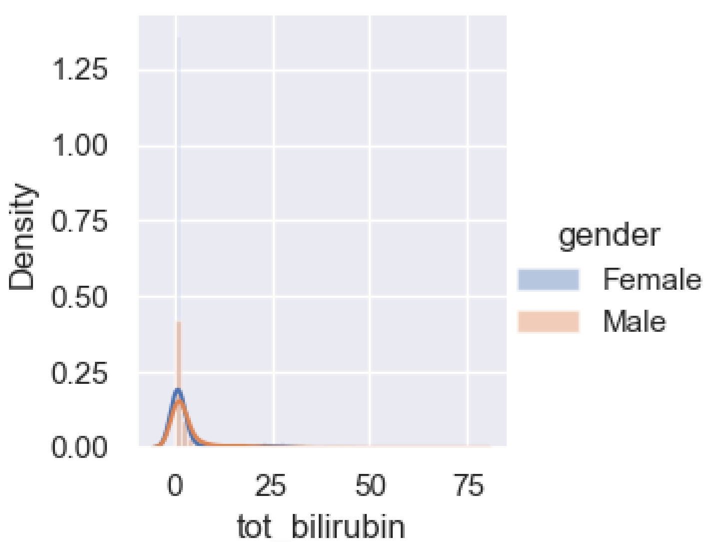
Histograms with Displot Plotting

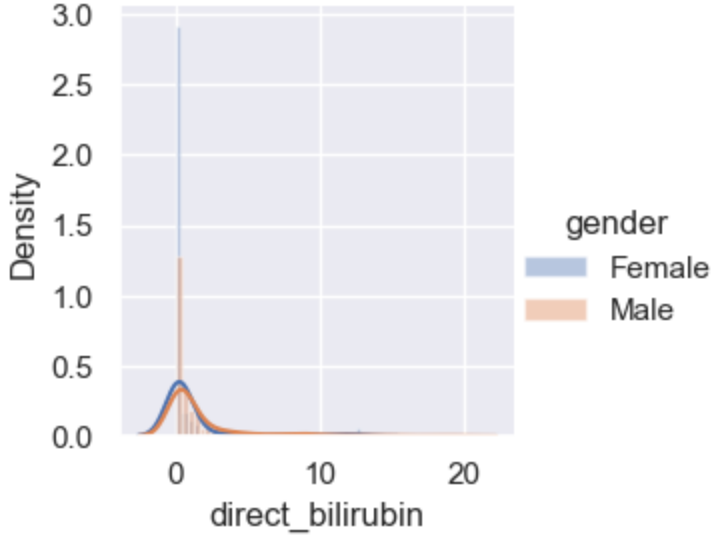
Displot is used basically for the univarient set of observations and vizualizes it through a histogram (i.e. only one observationand hence we choose one particular column of the dataset).

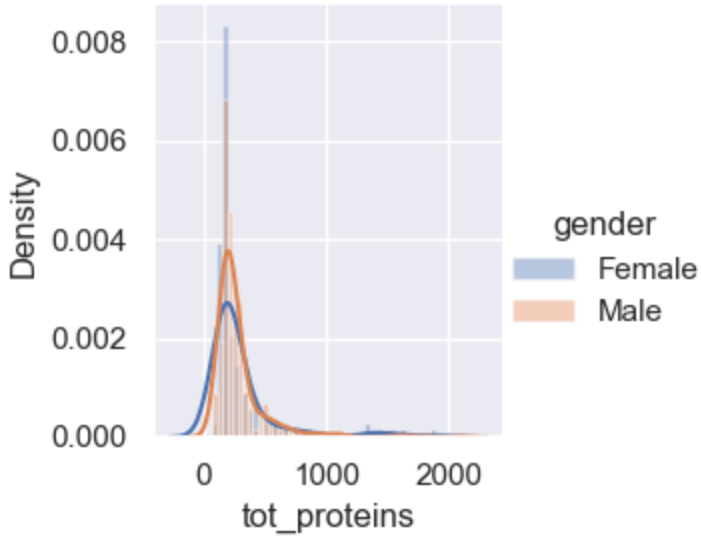
In [46]:
```
plot=sns.FacetGrid(df_liver,hue="gender")
plot.map(sns.distplot,"tot_bilirubin").add_legend()

plot=sns.FacetGrid(df_liver,hue="gender")
plot.map(sns.distplot,"direct_bilirubin").add_legend()

plot=sns.FacetGrid(df_liver,hue="gender")
plot.map(sns.distplot,"tot_proteins").add_legend()
```

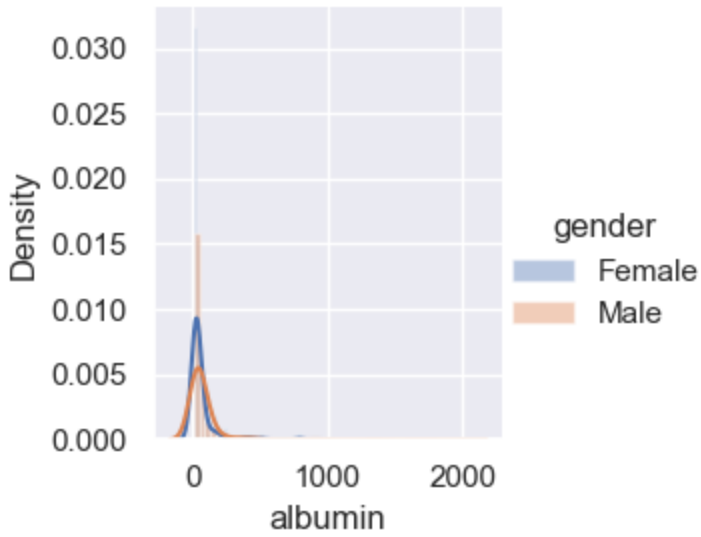
```
plot=sns.FacetGrid(df_liver,hue="gender")
plot.map(sns.distplot,"albumin").add_legend()

plt.show()
```

```
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/tarunsasanapuri/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.p
y:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```









Result Analysis:

From the above plots, we can see that-

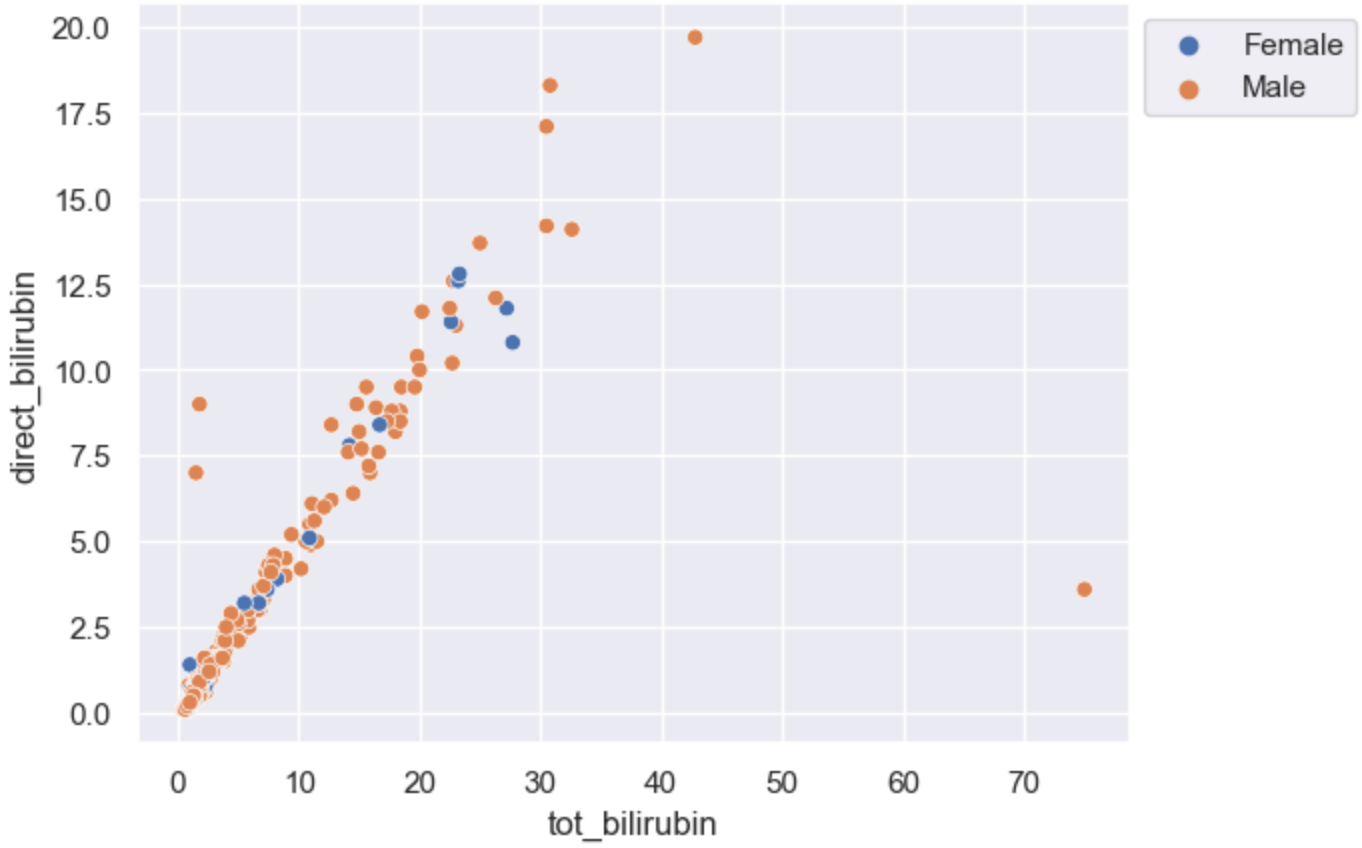
- In this case of Total Bilirubin,there is very little amount of overlapping
- In this case of Direct Bilirubin,there is little amount of overlapping
- In this case of Total Protein,there is huge amount of overlapping

- In this case of albumin,there is huge amount of overlapping

Scatter Plot

A scatter plot is a type of plot or mathematical diagram using cartesian coordinates to display values for typically two variables for a set of data

In [32]:
```
sns.scatterplot(x='tot_bilirubin',y='direct_bilirubin',hue='gender',data=df_liver)
plt.legend(bbox_to_anchor=(1,1),loc=2)
plt.show()
```

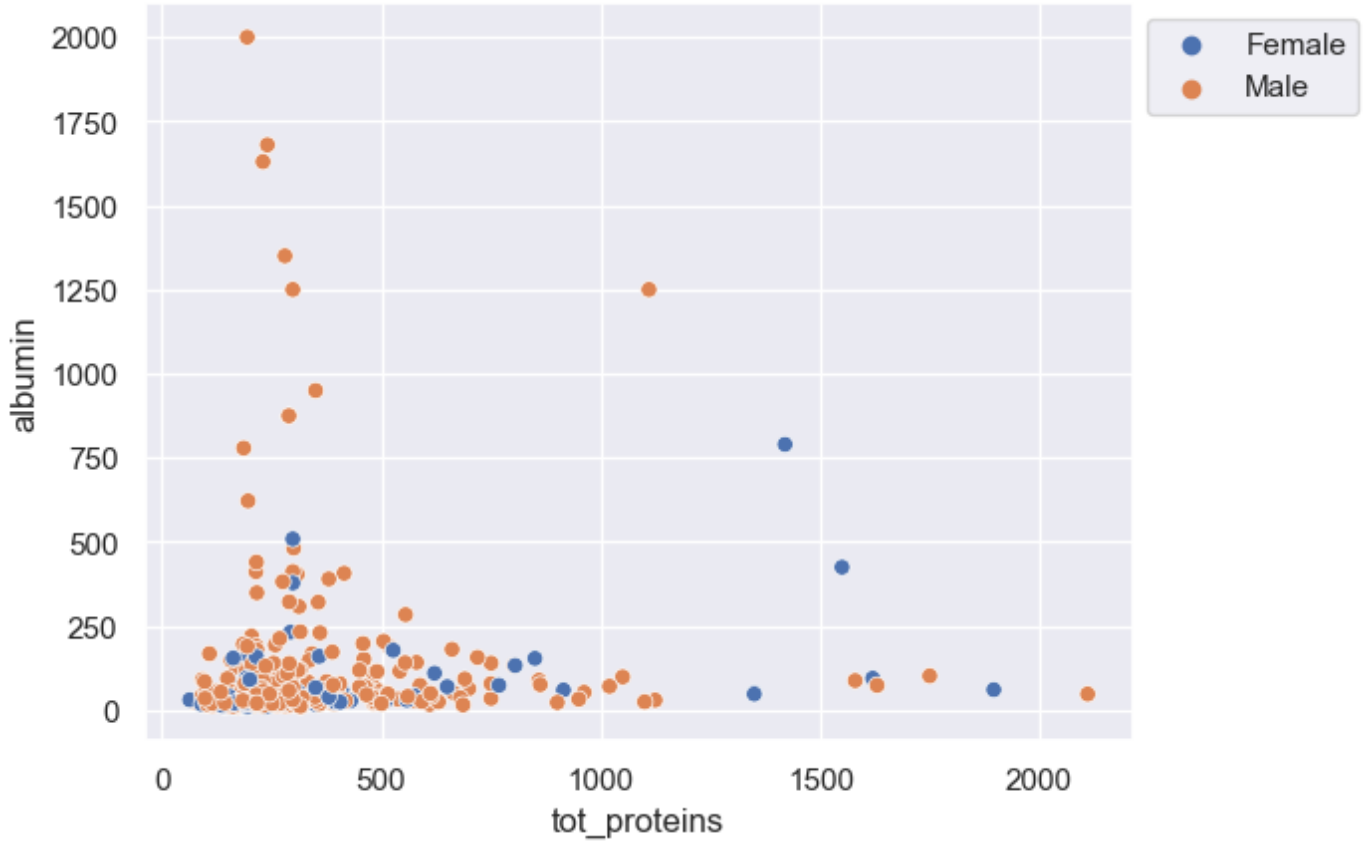


Result Analysis:

From the above plot,we can infer that-

- In gender, Male has larger tot_bilirubin as well as larger direct_bilirubin
- In gender, Female has smaller tot_bilirubin but not much larger direct_billirubin

In [33]:
```
sns.scatterplot(x='tot_proteins',y='albumin',hue='gender',data=df_liver)
# Placing legend outside the figure
plt.legend(bbox_to_anchor=(1,1),loc=2)
plt.show()
```



Result Analysis:

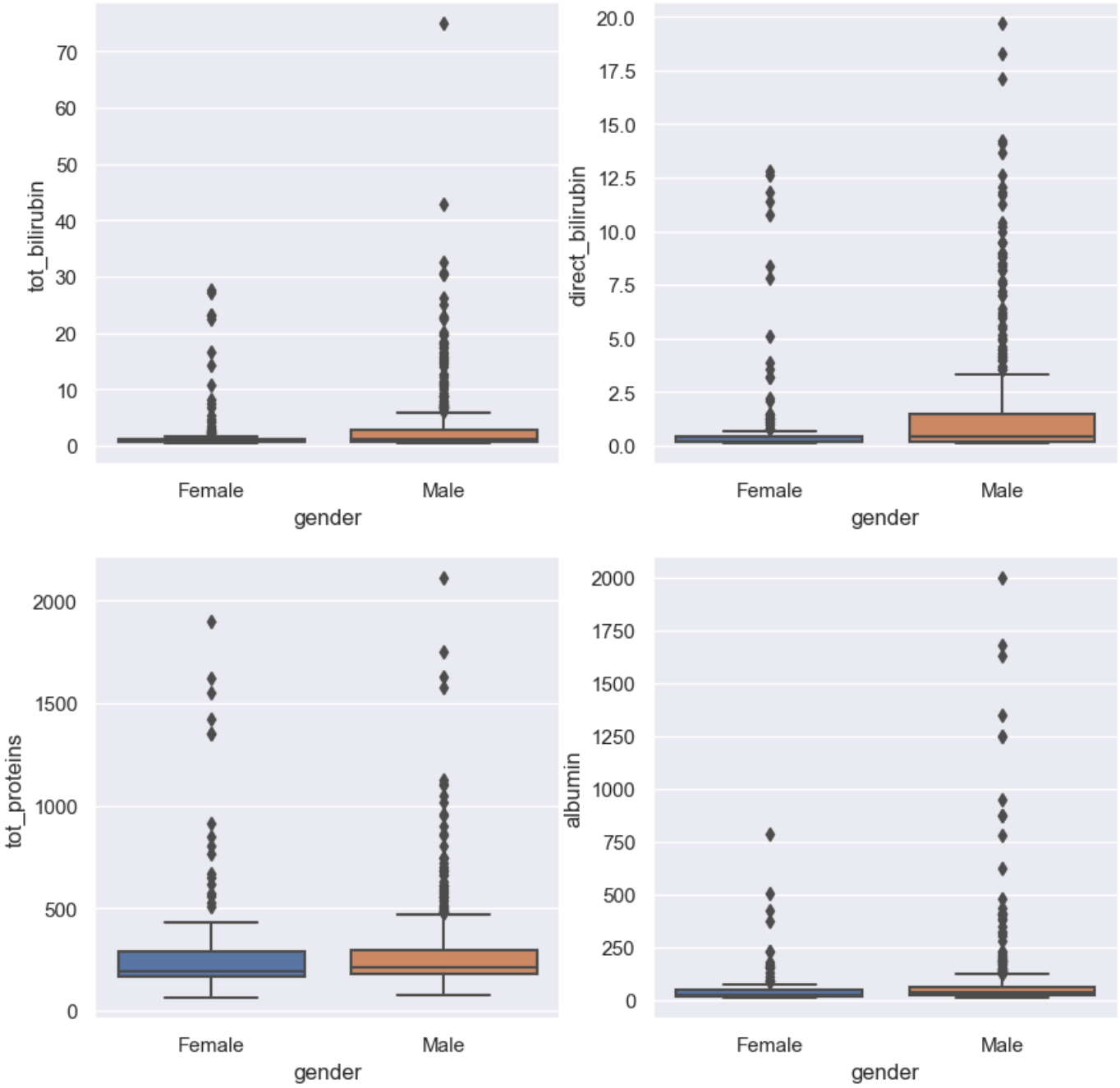
From the above plot,we can infer that-

- In gender,Male has larger Total_Proteins as well as highest Albumin
- In gender,Female has larger Total_Proteins but lesser albumin

Box Plot

We can use boxplots to see how the categorical value os distributed with other numerical values.

In [45]:
```
def graph(y):
    sns.boxplot(x='gender',y=y,data=df_liver)
plt.figure(figsize=(10,10))
plt.subplot(221)
graph('tot_bilirubin')

plt.subplot(222)
graph('direct_bilirubin')

plt.subplot(223)
graph('tot_proteins')

plt.subplot(224)
graph('albumin')

plt.show()
```



Result Analysis:

From the above graph,we can see that-

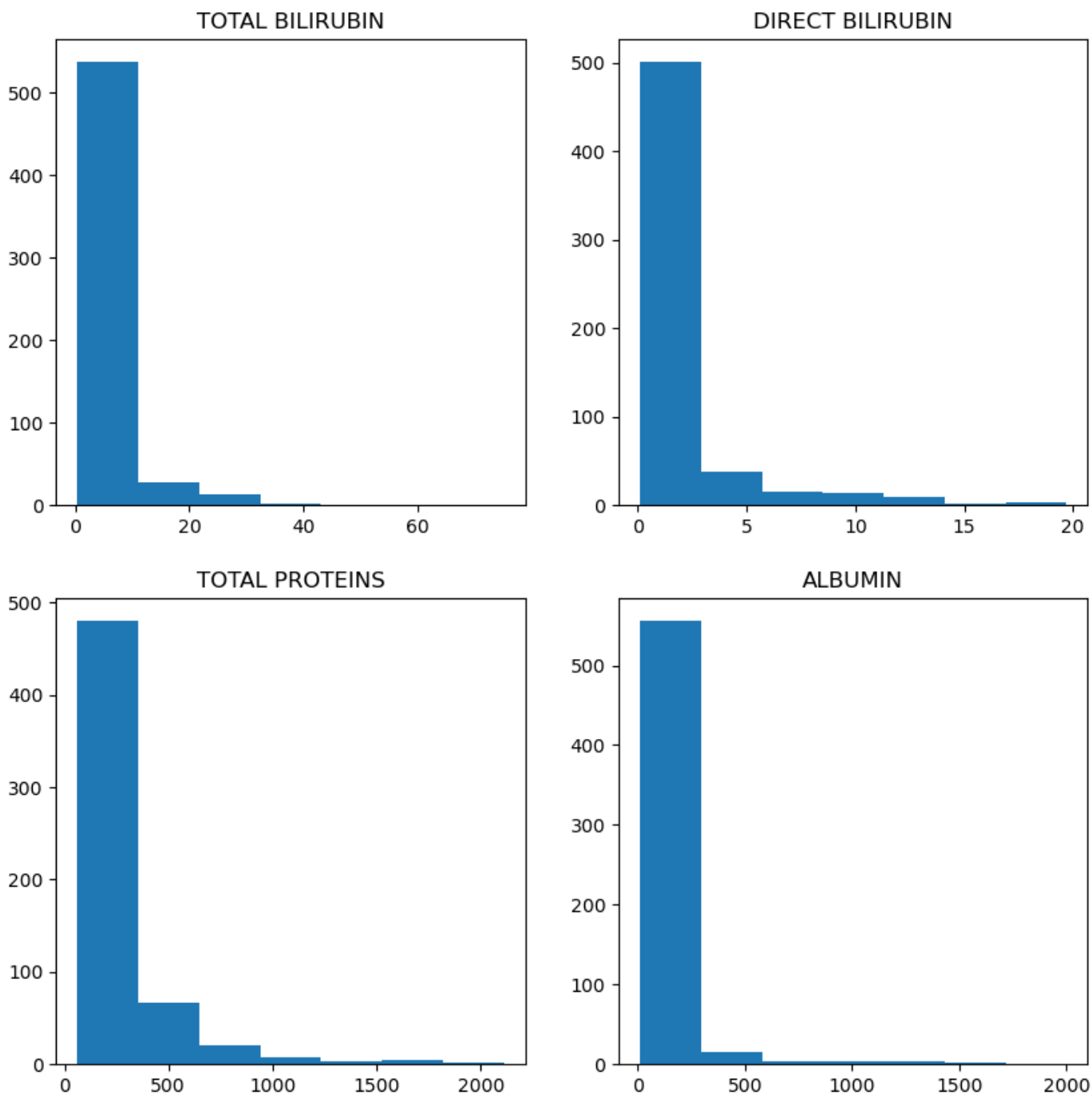
- In gender,Female has the smallest features and less distributed with some outliers
- In gender,Male has the highest features and more distributed with some outliers

Histograms

Histograms allow seeing the distribution of data for various columns. It can be used for uni as well as bi-variate analysis.

In [6]:
```
fig,axes=plt.subplots(2,2,figsize=(10,10))

axes[0,0].set_title("TOTAL BILIRUBIN")
axes[0,0].hist(df_liver['tot_bilirubin'],bins=7)

axes[0,1].set_title("DIRECT BILIRUBIN")
axes[0,1].hist(df_liver['direct_bilirubin'],bins=7)

axes[1,0].set_title("TOTAL PROTEINS")
axes[1,0].hist(df_liver['tot_proteins'],bins=7)

axes[1,1].set_title("ALBUMIN")
axes[1,1].hist(df_liver['albumin'],bins=7)
```

Out[6]:
```
(array([556.,  15.,   3.,   3.,   3.,   2.,   1.]),
 array([  10.        ,  294.28571429,  578.57142857,  862.85714286,
        1147.14285714, 1431.42857143, 1715.71428571, 2000.        ]),
 <BarContainer object of 7 artists>)
```



Result Analysis:

From the above plot,we can see that

- The highest frequency of Total_Bilirubin is above 500, which is between 0 and 20
- The highest frequency of Direct_Bilirubin is around 500 which is between 0 and 5
- The highest frequency of Total_Proteins is above 400,which is between 0 and 500
- The highest frequency of Albumin is above 500 which is between 0 and 500

Pair Plot

In [7]:
```
sns.pairplot(df_liver.drop(['age'],axis=1),hue='gender',height=2)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x7fe3b2f3b9d0>
```

Result Analysis

From the above plot,we can see that

- In every functions of the liver the ratio is highest for Male gender compare with Female

Heat Maps

The heatmap is a data visualization technique that is used to analyze the dataset as colors in two dimensions. Basically,it shows a correlation between all numerical variables in the dataset. In simpler terms, we can plot the above found correlation using the heatmaps

In [10]:
```
def correlation_heatmap(df):
    _ , ax = plt.subplots(figsize =(14, 12))
    colormap = sns.diverging_palette(220, 10, as_cmap = True)

    _ = sns.heatmap(
```

```
        df.corr(),
        cmap = colormap,
        square=True,
        cbar_kws={'shrink':.9 },
        ax=ax,
        annot=True,
        linewidths=0.1,vmax=1.0, linecolor='white',
        annot_kws={'fontsize':12 }
    )

    plt.title('Pearson Correlation of Features', y=1.05, size=15)

correlation_heatmap(df_liver)
```

Result Analysis:

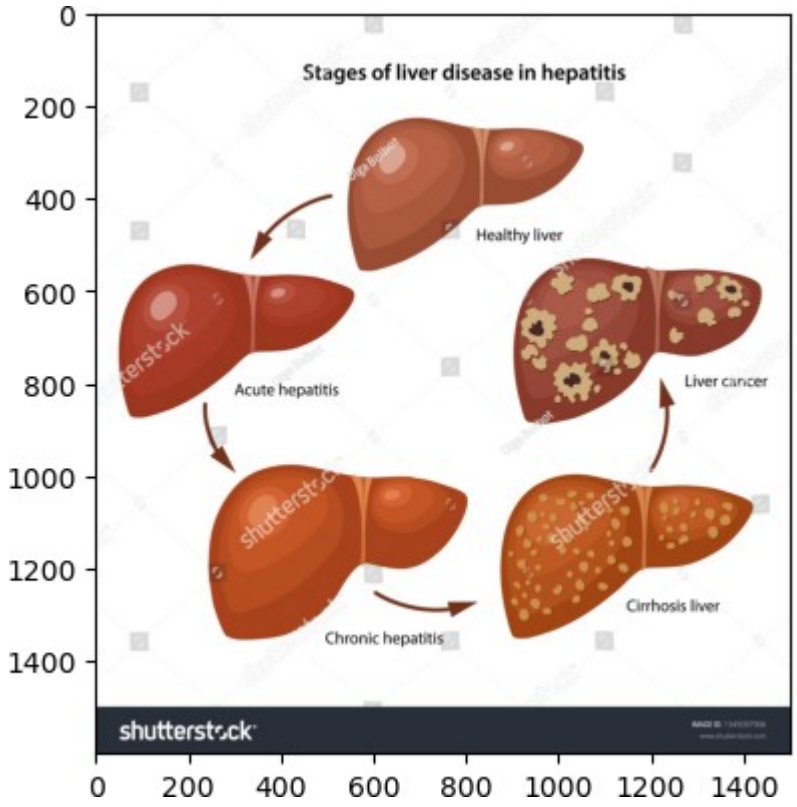
From the above plot,we can see that

- tot_bilirubin and direct_bilirubin has very high correlations.
- ag_ratio and albumin, sgpt and sgot, alkphos and sgot are having high correlations.
- sgot and age has very good correlations.

Image Visualization with Matplotlib

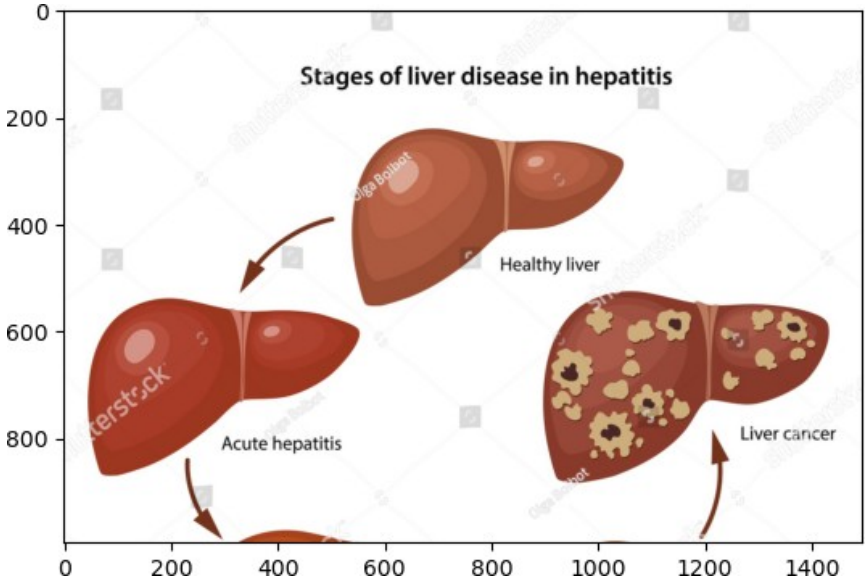
In [12]:
```
im=plt.imread("liverr.jpg")
plt.imshow(im)
```

Out[12]:
```
<matplotlib.image.AxesImage at 0x7fe3b3dfc7c0>
```



In [13]:
```
plt.imshow(im[5:1000,5:3000])
```

Out[13]:
```
<matplotlib.image.AxesImage at 0x7fe3c0312580>
```



Conclusion

The above dataset is about the liver disease dataset.This data set contains 416 liver patient records and167 non liver patient records.The data set was collected from test samples in North East of Andhra
Pradesh, India. 'is_patient' is a class label used to divide into groups(liver patient or not). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 islisted as being of age "90".

Acknowledgements

The data set has been elicit from UCI Machine Learning Repository. My sincere thanks to them.

Bibilography

Websites

www.kaggle.com(http://www.kaggle.com)
www.youtube.com(http://www.youtube.com)
www.wikipedia.com(http://www.wikipedia.com)