Finding the liver disease based on Classification of Indian Liver Patient Dataset

A PROJECT REPORT

for

SOFT COMPUTING TECHNIQUES (CSI3006)

in

Integrated M Tech (Computer Science and Engineering)

by

SAMITHA K(20MIC0099)

ABHINAYA B(20MIC0110)

THEJHASWINI R(20MIC0168)

Winter Semester, 2022-23

Under the Guidance of

Prof. AYYASAMY S

Professor, SCOPE

Submitted Journal: Soft Computing(Q2)

Submission ID: SOCO-D-23-02167

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

APRIL, 2023

DECLARATION BY THE CANDIDATE

We here by declare that the project report entitled **“Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computing technique”** submitted by us to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Soft Computing Techniques (CSI3006)** is a record of bonafide project work carried out by us under the guidance of **Prof. Ayyasamy S.** We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course.

Place : Vellore Signature

Date :

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitled **“Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computingtechnique”**submitted by **KSamitha(20MIC0099),B.Abhinaya(20MIC0110), Thejhaswini.R(20MIC0168)** to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Soft Computing Techniques (CSI3006)** is a record of bonafide work carried out by them undermy guidance.

Prof. Ayyasamy S
GUIDE
Professor, SCOPE

Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computing technique

S. Ayyasamy

[1] *Professor*, School of Computer Science and Engineering, *Vellore Institute of Technology, Tamil Nadu, India*

ayyasamy.s@vit.ac.in

B.Abhinaya

[2] *Post Graduate student*, School of Computer Science and Engineering, *Vellore Institute of Technology, Tamil Nadu, India*

abhinaya.bhimineni2020@vitstudent.ac.in

K.Samitha

[3] *Post Graduate of* School of Computer Science and Engineering, *Vellore Institute of Technology, TamilNadu, India*

kottam.samitha2020@vitstudent.ac.in

R.Thejashwini

[4] *Post Graduate of* School of Computer Science and Engineering, *Vellore Institute of Technology, TamilNadu, India*

thejashwini.r2020@vitstudent.ac.in

Abstract - This study uses a 'Logistic Regression', 'Gaussian Naive Bayes', 'Random Forest' method to try and achieve effective early diagnosis of liver illness. Using the UCI repository, we gathered 583 records pertaining to the Indian Liver Patient Dataset. 70% of the ILPD dataset is used for training, and 30% is used for testing. statistics of Indian liver patients Age, gender, total bilirubin, direct bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT, and alpha's are the 10 variables in this equation. As well as determining the overall accuracy, we will determine if the person has liver disease.

Keywords:

Precision, Accuracy, Regression, Liver disorder, Recall

Introduction:

The liver controls a number of potentially harmful bodily processes, and if it develops a disease or is destroyed, the body may suffer serious harm as a result of the lack of those processes. Hepatic disease is another name for liver disease. The broad phrase "liver disease" refers to all possible issues that could prevent the liver from carrying out its intended duties. Typically, three quarters or more of the liver's tissue must be damaged before liver function starts to decline.

This paper describes the approach, one of the most used supervised classification methods. The use of classification systems in various automatic medical diagnostics is very common. While the liver will continue to operate correctly even when it is partially damaged, problems with liver patients are difficult to identify at an early stage. The likelihood that a patient will survive will rise with an early diagnosis of liver issues. Enzyme levels in the blood can be analysed to diagnose liver disease. Age, gender, total bilirubin, direct bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT, and Alpo's are the 10 variables in the Indian liver patient dataset T.

Nowadays, medical professionals frequently employ artificial intelligence to detect a variety of illnesses that are brought on by the dysfunction of particular organs.

Literature survey :

Authors	Methodology or Techniques used	Advantages	Issues	Metrics used
1)Jeddah	Genetic algorithms, Computer assisted diagnosis	improve the efficiency and effectiveness of the admission process	The improved method avoids computing the distance of each data object to the cluster centers repeatly, saving the running time	Supervised learning technique
2)Himani Sharma	ANN, Back propagation diagnosis, Feed Forward Neural Network	Accuracy was increased by 1%.	This population also appears to be predisposed to developing this disease earlier, compared to the Western population	Decision tree
3)Mr. Brijain R Patel, Mr. Kushik K	Decision tree, Back propagation Neural Network	Accuracy was increased by 2%	focus on the various algorithms	Classification and prediction are the techniques used to make out important data classes and predict probable

4)Huang Ming	Data mining classification, Neural Networks, Parallelism	simplifies the information entropy solution of ID3 algorithm	Alcoholic liver disease (ALD) is one of the main causes of chronic liver disease worldwide	It accounts for up to 48% of cirrhosis-associated deaths
5) Niu Wenying	Back propogation networks, Genetic algorithm	accracy was increased by 5%	It has been reported that haemodialysis increases the possibility of blood borne viral infection but the prevalence is variable from haemodialysis from centre to centre and also from region to region and country to country, and high-cost haemodialysis centre vs low-cost haemodialysis centre.	In most of the study, HBV infection among hemodialysis patient was between 4 and 11%
6) Vaidya, M.HChaudri	Artificial neural Networks, Fuzzy logic, Fuzzy Neural Network, Classification,	Early diagnosis is of considerable amount of significance in treating the disease. Diagnosis is of the physician skills conducting based on their knowledge’s and experience yet an error might occurrence is here	It cannot be a lot of possible errors in this diagnosis due to the number of enzymes to be many as well as the effects of different taken alcohol rates to be very from one patient to the other.	The Liver Disorders includes 345 specimens consisting of six fields and two classes. Each sample is taken from an single man. Two hundred of these samples are of one class with remaining 145 are possessed by to the other.
7) Vijayarani.s Dhayanand.s	Artificial Neural Network (ANN) classification algorithm. LS-	This dataset contains Liver Function Test details (LFT).	Utilized PC and LSSVM doesn’t give the expected results	Diabetes Dataset Indian Liver Patient Dataset (ILPD). Dataset contains Liver Function Test details (LFT).

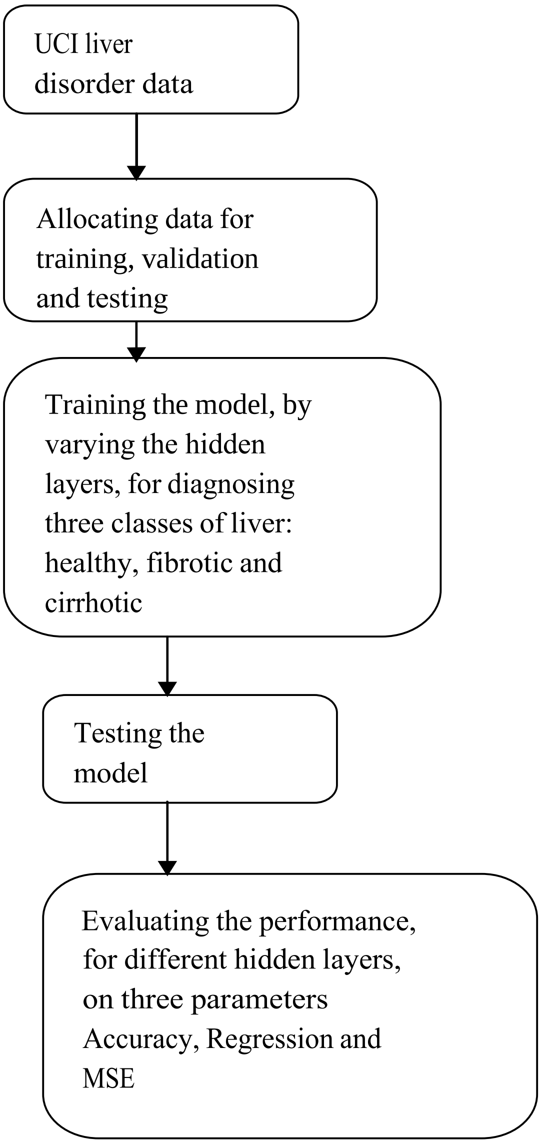
	SVM algorithm	Karthik et.al were applied a soft computing technique for intelligent diagnosis of liver disease. They have implemented classification and its type detection in phases.		
8) Lin R.H	Random forest algorithm, classification, computational intelligence,	It is shown that feature selection has a great significance as the process of selecting a subset of relevant features for use in model construction. By using feature selection on ILPD before a classification algorithm can be applied, performance of classification algorithm increases.	Problems with liver patients are not easily discovered in an early stage as it will be functioning normally even when it is partially damaged [2]. An early diagnosis of liver problems will increase patient’s survival rate.	Classifying Banking Dataset, Indian Liver Patient Dataset(ILDP)
9) Jankisharan Pahareeya Rajan Vohra Jagdish Makhijani Sanjay Patsariya	Multilayer Feed Forward Neural Network, Random Forest, Multiple Linear Regression (MLR), Support Vector Machine (SVM) and Genetic programming (GP).	The results indicates that there exists more significant difference in the groups with all the possible attribute combinations except analysis on SGPT between non liver patients of UCI and INDIA data sets	the accuracy of these models is not satisfactory so there is always a scope for new classifactory models.	ILPD data set and UCI data set
10) Kalyan	Discriminative	To serve the	Identification of	It was followed by splitting of

Nagaraj and Amulyashree Sridhar	learning, Artificial Neural Network, Bagging, Boosting, Naïve Bayes, Kernel-based classifiers, Nearest Neighbour algorithm, Decision Trees, Random Forest,	medicinal community for prediction of liver disease among patients, a graphical user interface (GUI) has been developed using R. The GUI is deployed as a package in local repository of R platform for users to perform prediction.	liver infection at preliminary stage is important but combat the frequency and severity deaths of patients in India are higher. The patients must be screened based on initial symptoms for development of personalized therapy.	the dataset into training (70% of the dataset) and test (30%) sets. Training set comprised of 389 instances and test set included the remaining 194 instances.

PROPOSED WORK :

ARCHITECTURE :

Fig-1


UCI liver disorder data
Allocating data for training, validation and testing
Training the model, by varying the hidden layers, for diagnosing three classes of liver: healthy, fibrotic and cirrhotic
Testing the model
Evaluating the performance, for different hidden layers, on three parameters Accuracy, Regression and MSE

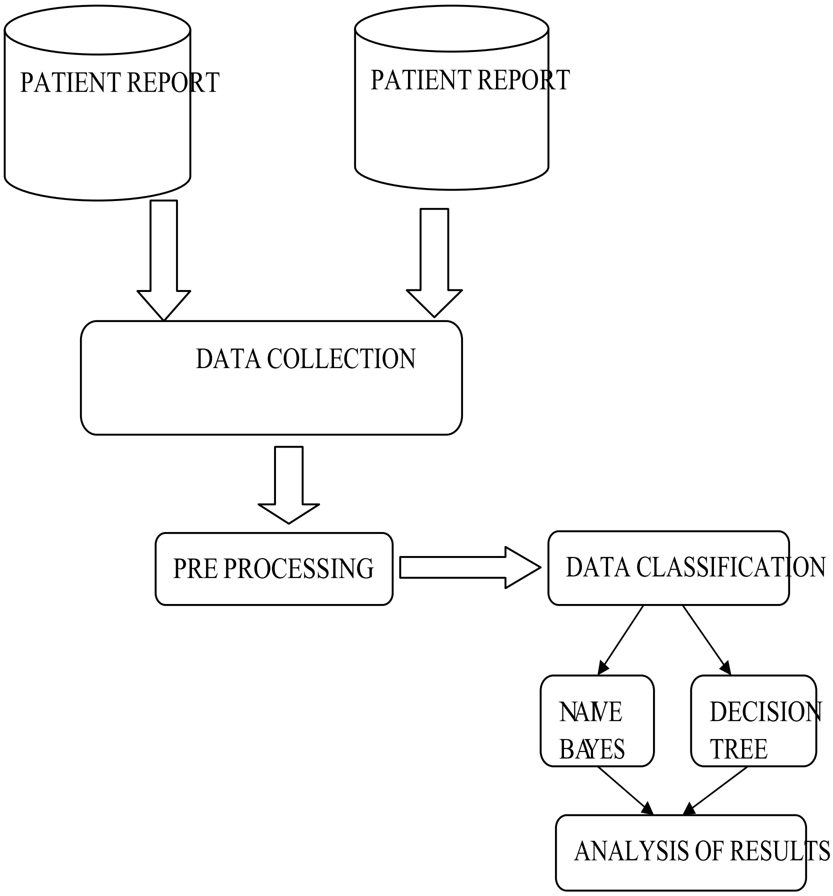


Fig-2

DECISION TREE :

Using Euclidean distance similarity, divide the training cases into k clusters. We construct decision trees using the C4.5 decision tree technique on each cluster, which represents a density region of typical or anomalous cases.

PRE PROCESSING:

Characteristics must range from 0 to 1. The action is known as normalisation. Each sample of a particular property is normalised by dividing it by its greatest value.

Gaussian Naive Bayes

When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Sometimes assume variance

- is independent of Y (i.e., σi)
- or independent of Xi (i.e., σk)
- or both (i.e., σ)

PATIENT REPORT:

The patient report is very important.Since the patient has access to all information regarding their diagnosis, medical history, prescriptions, and appointment times. It must not be mixed up with any other patient

Evaluation Metrics Used -

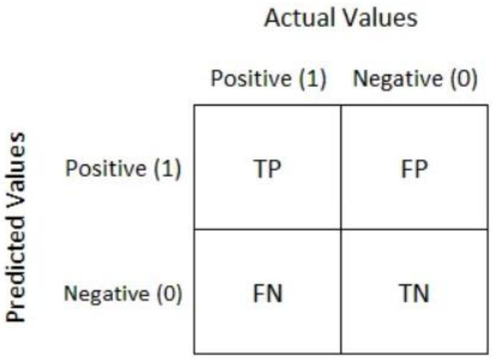
Since this is binary classification problem, we use the following metrics:

- Confusion matrix - For getting a better clarity of the no of correct/incorrect predictions by the model.

In order for the classifier to work at its best, the attribute values must be converted into homogenous, well-behaved values that generate numerical stability. As a result, the values of the   patients. They must very carefully safeguard the patient data. It shouldn't be in a risky situation. Data gathering is a crucial procedure. Data shouldn't be mixed up with patient information

DATA COLLECTION: Here Data is collected and we perform the required methods.

The Indian Liver Patient Dataset collects patient data, which is then stored in several databases. They collect the data, analyse it, and then communicate the findings to the information. Moreover, it never exchanges by error. The patient report must always be given to the appropriate patients.



Actual Values

Predicted Values

	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Confusion Metrics

From our confusion matrix, we can calculate five

different metrics measuring the validity of our model.

1. Accuracy (all **correct** / all)

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

2. Misclassification (all **incorrect** / all) = FP + FN / TP + TN + FP + FN

3. Precision (**true** positives / **predicted** positives) =

$$Precision = \frac{TP}{TP + FP}$$

Sensitivity aka Recall (**true** positives / all **actual** positives) =

$$Recall = \frac{TP}{TP + FN}$$

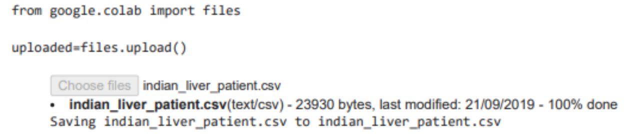
Specificity (**true** negatives / all **actual** negatives) =TN / TN + FP

4) F1 score

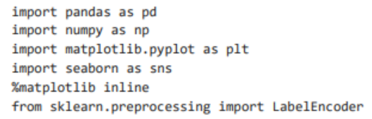
$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

EXPERIMENTS AND RESULTS:

Analysis and prediction of Indian liver patient

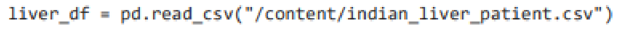


```
from google.colab import files

uploaded=files.upload()

    Choose files indian_liver_patient.csv
    • indian_liver_patient.csv(text/csv) - 23930 bytes, last modified: 21/09/2019 - 100% done
    Saving indian_liver_patient.csv to indian_liver_patient.csv
```

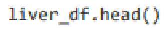


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import LabelEncoder
```

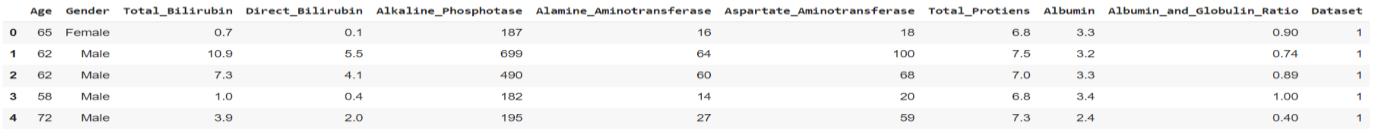
Data Analysis:



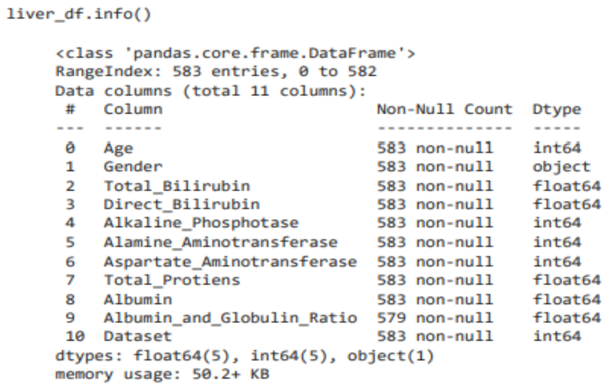
```
liver_df = pd.read_csv("/content/indian_liver_patient.csv")
```



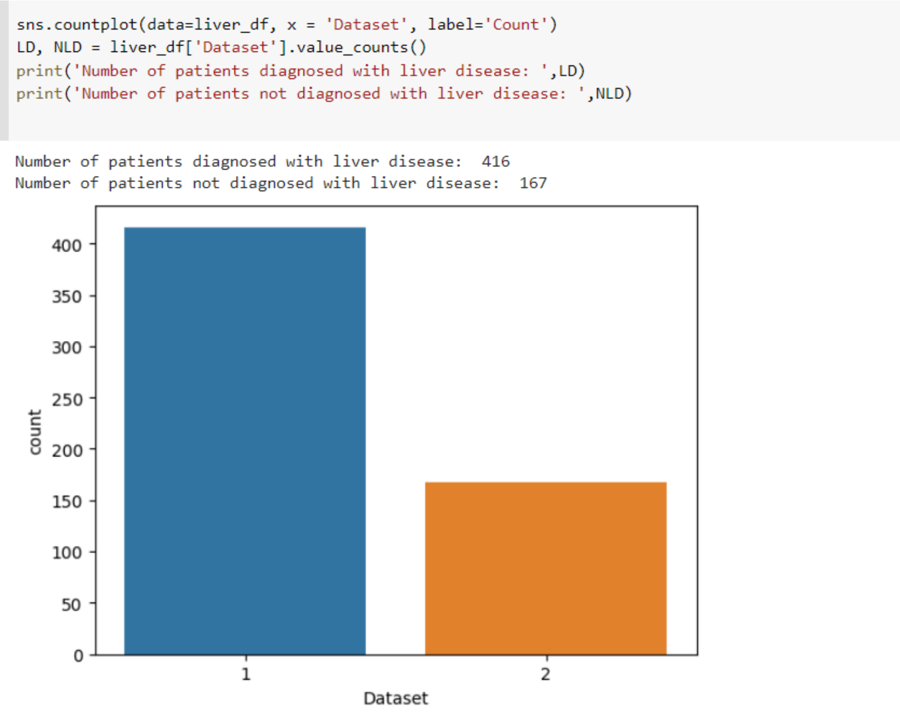
```
liver_df.head()
```



	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
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



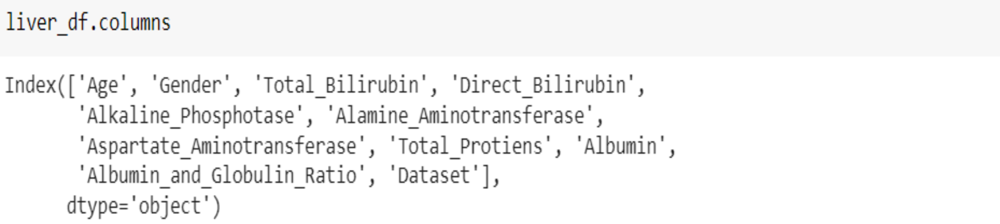
```
liver_df.info()

      <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   Alamine_Aminotransferase    583 non-null    int64
       6   Aspartate_Aminotransferase  583 non-null    int64
       7   Total_Protiens              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: 50.2+ KB
```



```
sns.countplot(data=liver_df, x = 'Dataset', label='Count')
LD, NLD = liver_df['Dataset'].value_counts()
print('Number of patients diagnosed with liver disease: ',LD)
print('Number of patients not diagnosed with liver disease: ',NLD)
```

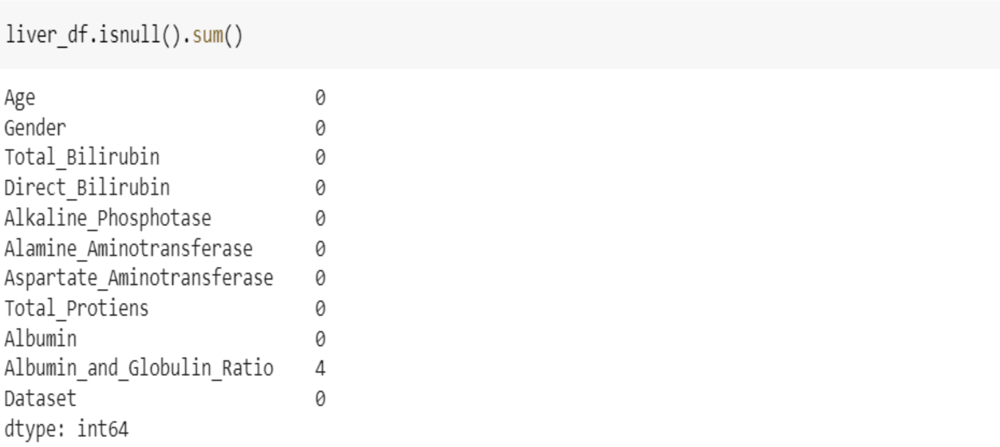
```
Number of patients diagnosed with liver disease:  416
Number of patients not diagnosed with liver disease:  167
```

Fig-3



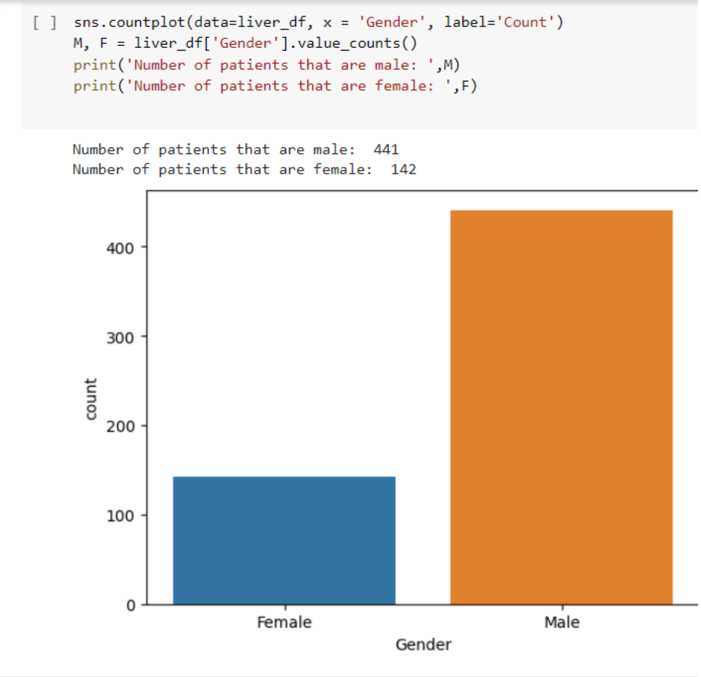
```
liver_df.columns
```

```
Index(['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
       'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
       'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
       'Albumin_and_Globulin_Ratio', 'Dataset'],
      dtype='object')
```



```
liver_df.isnull().sum()
```

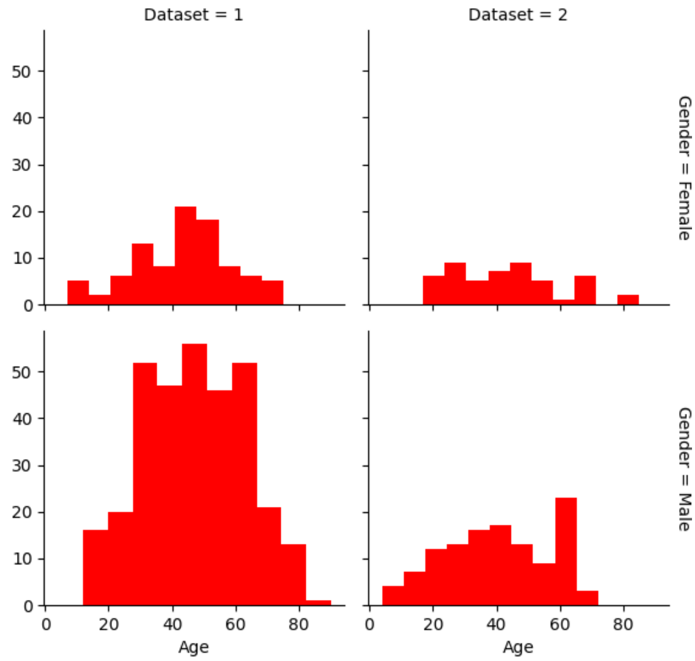
```
Age                           0
Gender                        0
Total_Bilirubin               0
Direct_Bilirubin              0
Alkaline_Phosphotase          0
Alamine_Aminotransferase      0
Aspartate_Aminotransferase    0
Total_Protiens                0
Albumin                       0
Albumin_and_Globulin_Ratio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

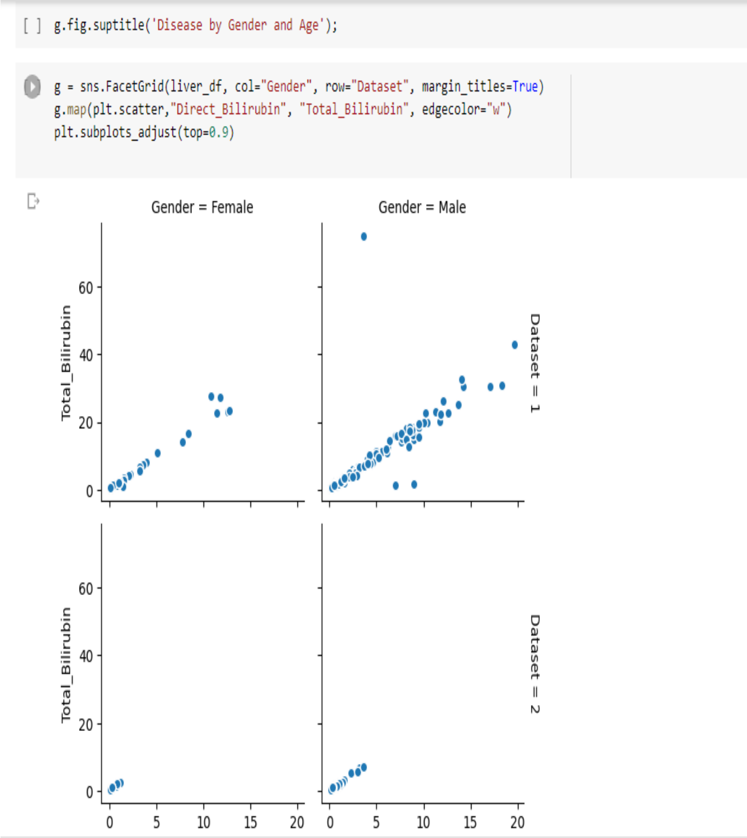


```
[ ] sns.countplot(data=liver_df, x = 'Gender', label='Count')
    M, F = liver_df['Gender'].value_counts()
    print('Number of patients that are male: ',M)
    print('Number of patients that are female: ',F)
```

```
Number of patients that are male:  441
Number of patients that are female:  142
```

Fig-4

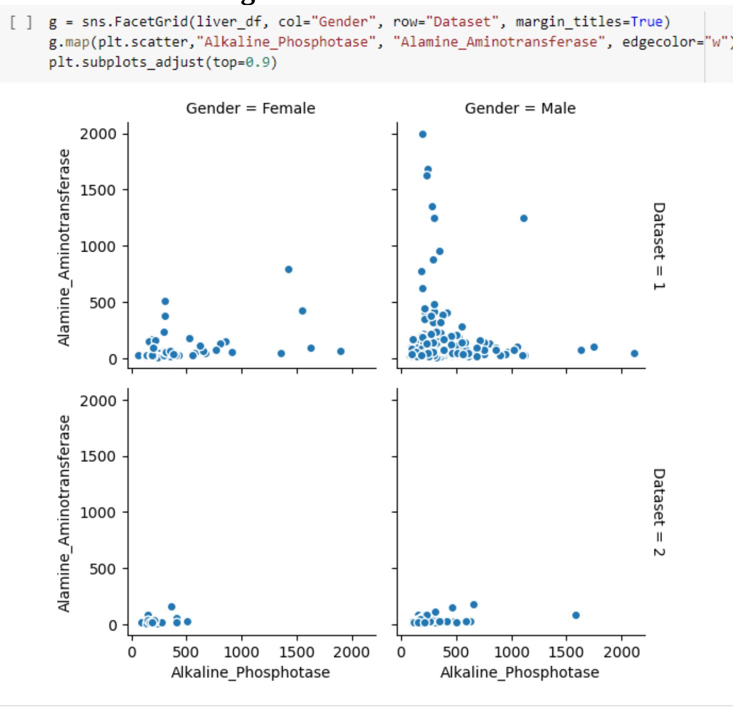




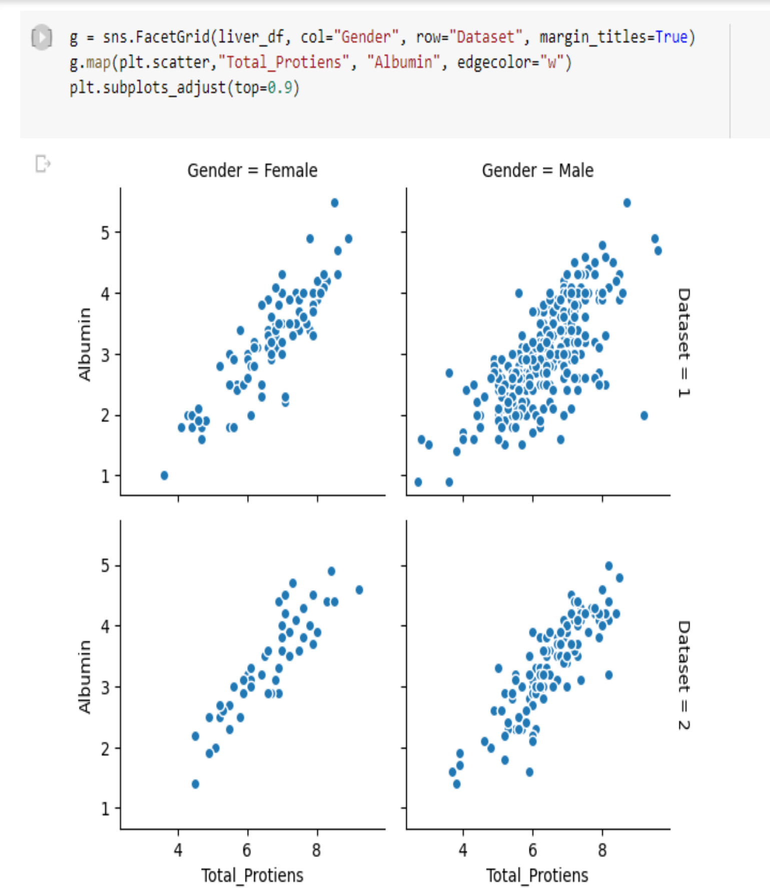
```
[ ] g.fig.suptitle('Disease by Gender and Age');
```

```
g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
g.map(plt.scatter,"Direct_Bilirubin", "Total_Bilirubin", edgecolor="w")
plt.subplots_adjust(top=0.9)
```

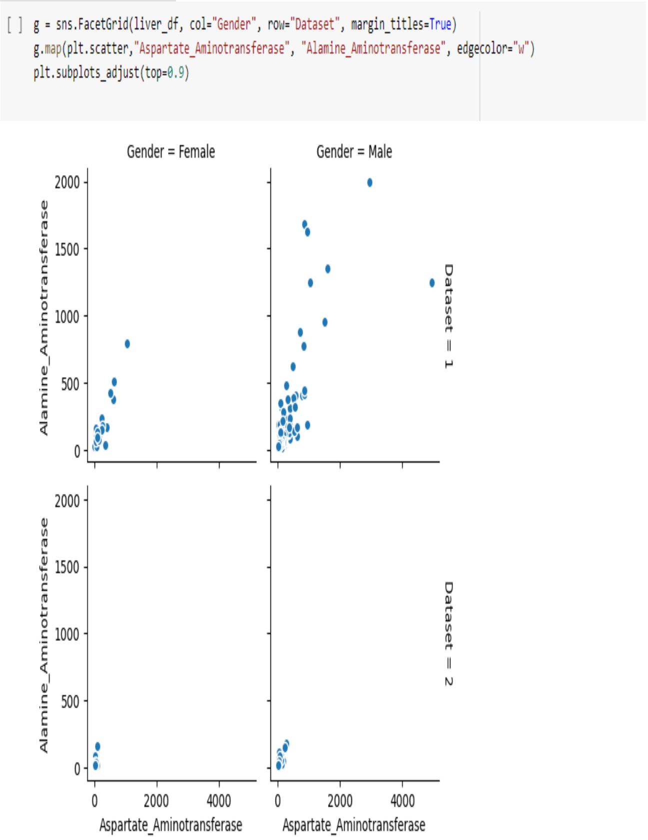
Fig-5



```
[ ] g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
    g.map(plt.scatter,"Alkaline_Phosphotase", "Alamine_Aminotransferase", edgecolor="w"
    plt.subplots_adjust(top=0.9)
```



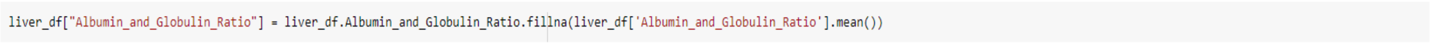
```
g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
g.map(plt.scatter,"Total_Protiens", "Albumin", edgecolor="w")
plt.subplots_adjust(top=0.9)
```



```
[ ] g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
    g.map(plt.scatter,"Aspartate_Aminotransferase", "Alamine_Aminotransferase", edgecolor="w")
    plt.subplots_adjust(top=0.9)
```

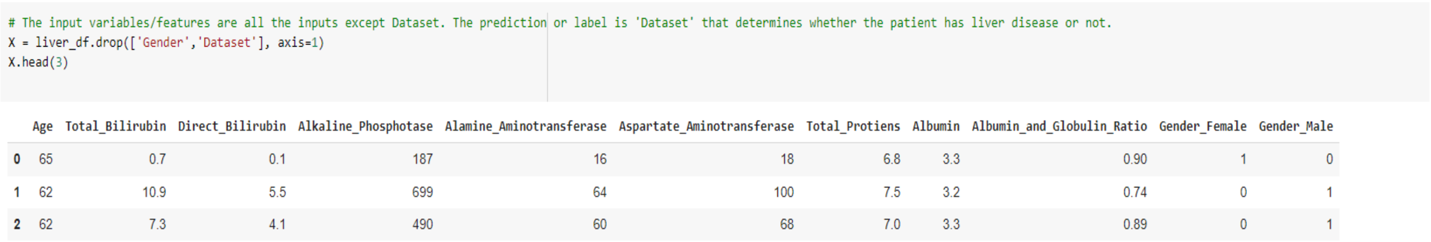
Fig-6

Fig-7



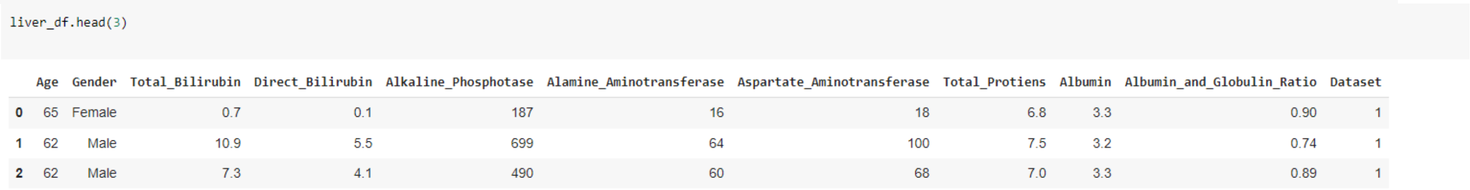
```
liver_df["Albumin_and_Globulin_Ratio"] = liver_df.Albumin_and_Globulin_Ratio.fillna(liver_df['Albumin_and_Globulin_Ratio'].mean())
```

```
#liver_df[liver_df['Albumin_and_Globulin_Ratio'] == 0.9470639032815201]
```



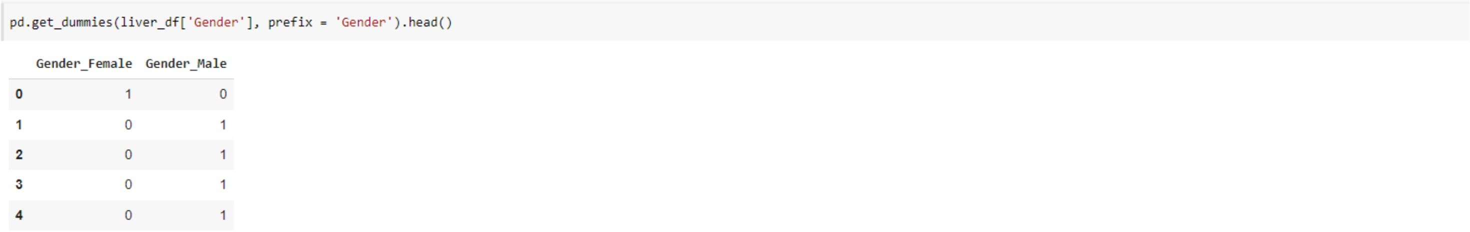
```
# The input variables/features are all the inputs except Dataset. The prediction or label is 'Dataset' that determines whether the patient has liver disease or not.
X = liver_df.drop(['Gender','Dataset'], axis=1)
X.head(3)
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Gender_Female	Gender_Male
0	65	0.7	0.1	187	16	18	6.8	3.3	0.90	1	0
1	62	10.9	5.5	699	64	100	7.5	3.2	0.74	0	1
2	62	7.3	4.1	490	60	68	7.0	3.3	0.89	0	1



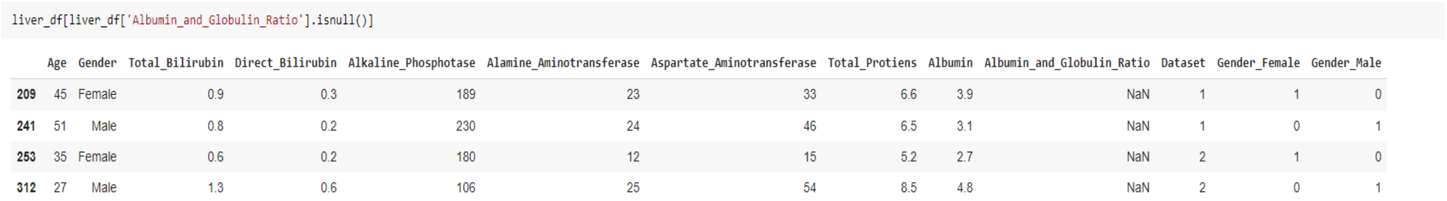
```
liver_df.head(3)
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
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



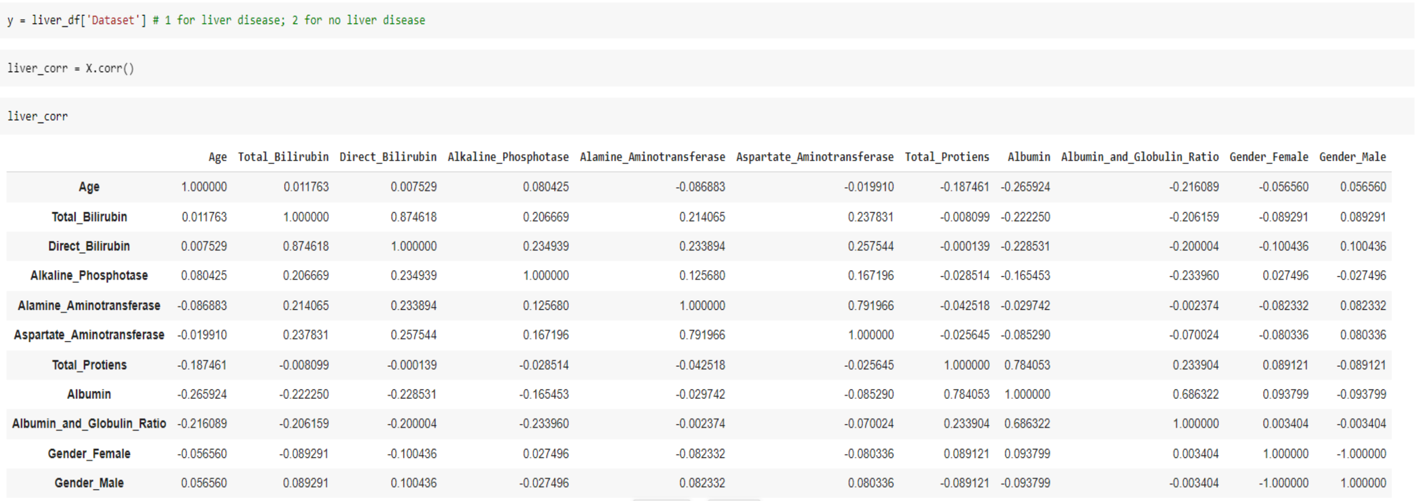
```
pd.get_dummies(liver_df['Gender'], prefix = 'Gender').head()
```

	Gender_Female	Gender_Male
0	1	0
1	0	1
2	0	1
3	0	1
4	0	1



```
liver_df[liver_df['Albumin_and_Globulin_Ratio'].isnull()]
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset	Gender_Female	Gender_Male
209	45	Female	0.9	0.3	189	23	33	6.6	3.9	NaN	1	1	0
241	51	Male	0.8	0.2	230	24	46	6.5	3.1	NaN	1	0	1
253	35	Female	0.6	0.2	180	12	15	5.2	2.7	NaN	2	1	0
312	27	Male	1.3	0.6	106	25	54	8.5	4.8	NaN	2	0	1

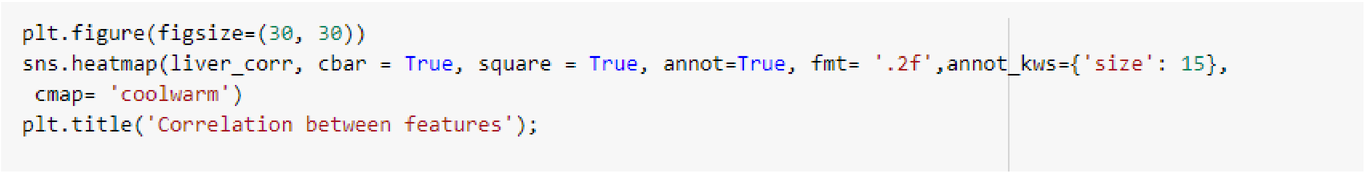


```
y = liver_df['Dataset'] # 1 for liver disease; 2 for no liver disease
```

```
liver_corr = X.corr()
```

```
liver_corr
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Gender_Female	Gender_Male
Age	1.000000	0.011763	0.007529	0.080425	-0.086883	-0.019910	-0.187461	-0.265924	-0.216089	-0.056560	0.056560
Total_Bilirubin	0.011763	1.000000	0.874618	0.206669	0.214065	0.237831	-0.008099	-0.222250	-0.206159	-0.089291	0.089291
Direct_Bilirubin	0.007529	0.874618	1.000000	0.234939	0.233894	0.257544	-0.000139	-0.228531	-0.200004	-0.100436	0.100436
Alkaline_Phosphotase	0.080425	0.206669	0.234939	1.000000	0.125680	0.167196	-0.028514	-0.165453	-0.233960	0.027496	-0.027496
Alamine_Aminotransferase	-0.086883	0.214065	0.233894	0.125680	1.000000	0.791966	-0.042518	-0.029742	-0.002374	-0.082332	0.082332
Aspartate_Aminotransferase	-0.019910	0.237831	0.257544	0.167196	0.791966	1.000000	-0.025645	-0.085290	-0.070024	-0.080336	0.080336
Total_Protiens	-0.187461	-0.008099	-0.000139	-0.028514	-0.042518	-0.025645	1.000000	0.784053	0.233904	0.089121	-0.089121
Albumin	-0.265924	-0.222250	-0.228531	-0.165453	-0.029742	-0.085290	0.784053	1.000000	0.686322	0.093799	-0.093799
Albumin_and_Globulin_Ratio	-0.216089	-0.206159	-0.200004	-0.233960	-0.002374	-0.070024	0.233904	0.686322	1.000000	0.003404	-0.003404
Gender_Female	-0.056560	-0.089291	-0.100436	0.027496	-0.082332	-0.080336	0.089121	0.093799	0.003404	1.000000	-1.000000
Gender_Male	0.056560	0.089291	0.100436	-0.027496	0.082332	0.080336	-0.089121	-0.093799	-0.003404	-1.000000	1.000000



```
plt.figure(figsize=(30, 30))
sns.heatmap(liver_corr, cbar = True, square = True, annot=True, fmt= '.2f',annot_kws={'size': 15},
 cmap= 'coolwarm')
plt.title('Correlation between features');
```

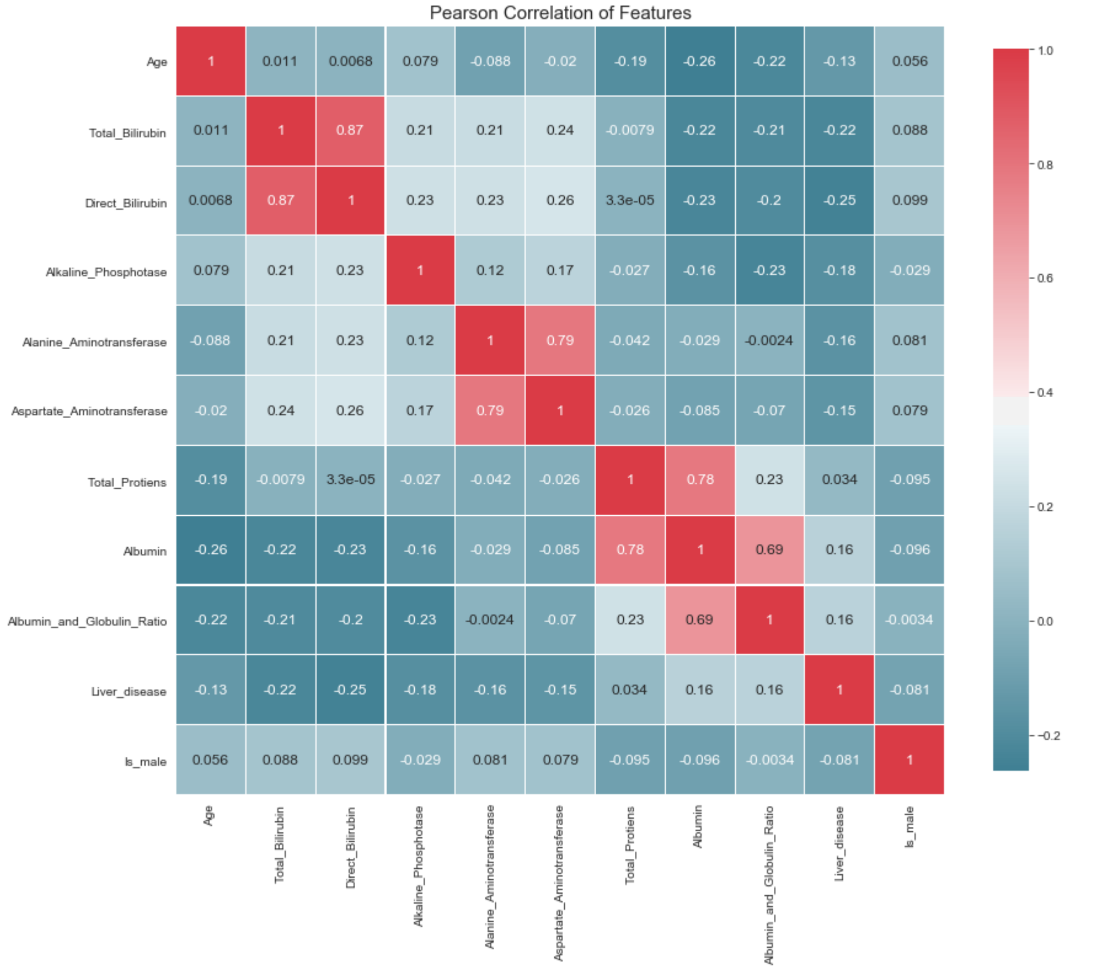
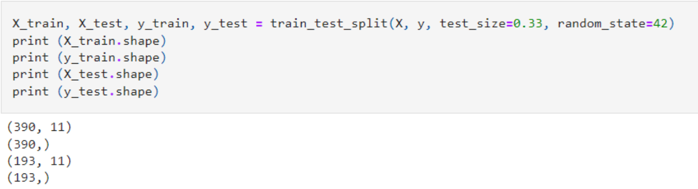


Fig-8

Splitting the data into Train and Test

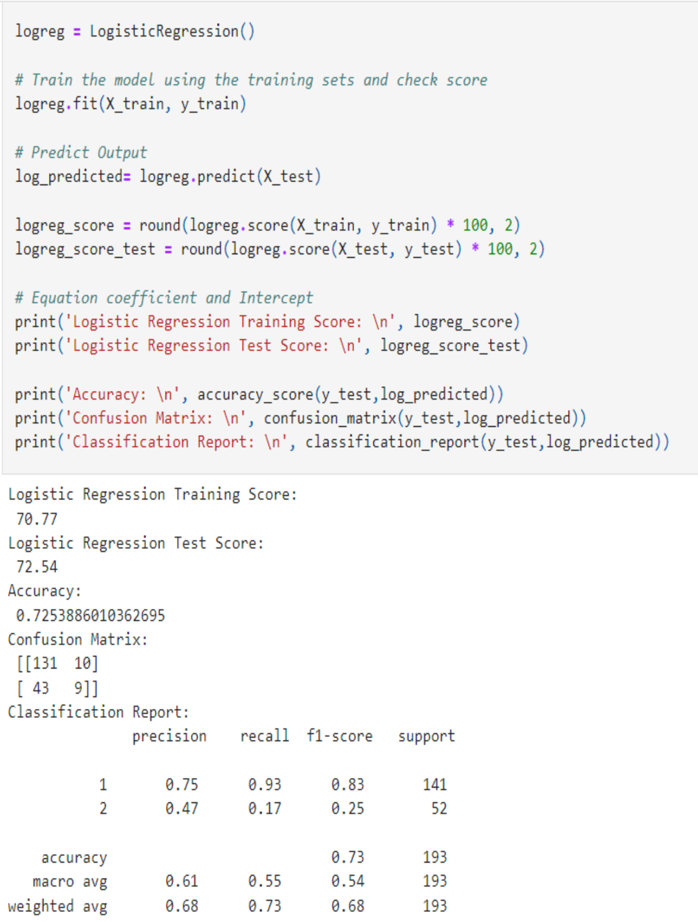


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
print (X_train.shape)
print (y_train.shape)
print (X_test.shape)
print (y_test.shape)
```

```
(390, 11)
(390,)
(193, 11)
(193,)
```

Model Building

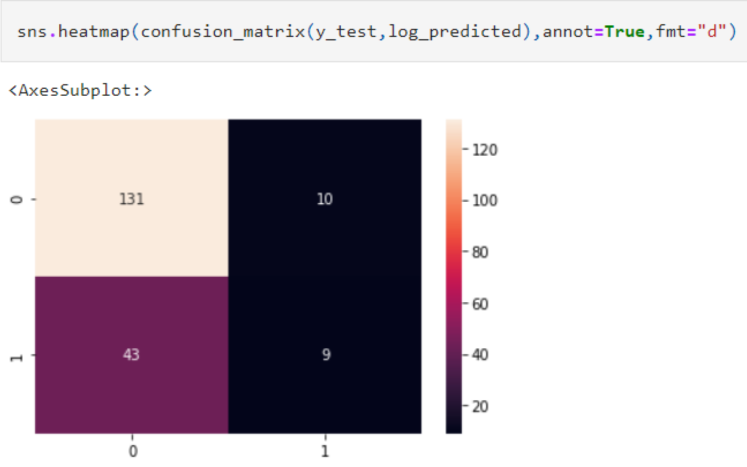
1. Logistic Regression



```
logreg = LogisticRegression()

# Train the model using the training sets and check score
logreg.fit(X_train, y_train)

# Predict Output
log_predicted= logreg.predict(X_test)

logreg_score = round(logreg.score(X_train, y_train) * 100, 2)
logreg_score_test = round(logreg.score(X_test, y_test) * 100, 2)

# Equation coefficient and Intercept
print('Logistic Regression Training Score: \n', logreg_score)
print('Logistic Regression Test Score: \n', logreg_score_test)

print('Accuracy: \n', accuracy_score(y_test,log_predicted))
print('Confusion Matrix: \n', confusion_matrix(y_test,log_predicted))
print('Classification Report: \n', classification_report(y_test,log_predicted))
```

```
Logistic Regression Training Score: 
 70.77
Logistic Regression Test Score: 
 72.54
Accuracy: 
 0.7253886010362695
Confusion Matrix: 
 [[131  10]
 [ 43   9]]
Classification Report: 
               precision    recall  f1-score   support

           1       0.75      0.93      0.83       141
           2       0.47      0.17      0.25        52

    accuracy                           0.73       193
   macro avg       0.61      0.55      0.54       193
weighted avg       0.68      0.73      0.68       193
```

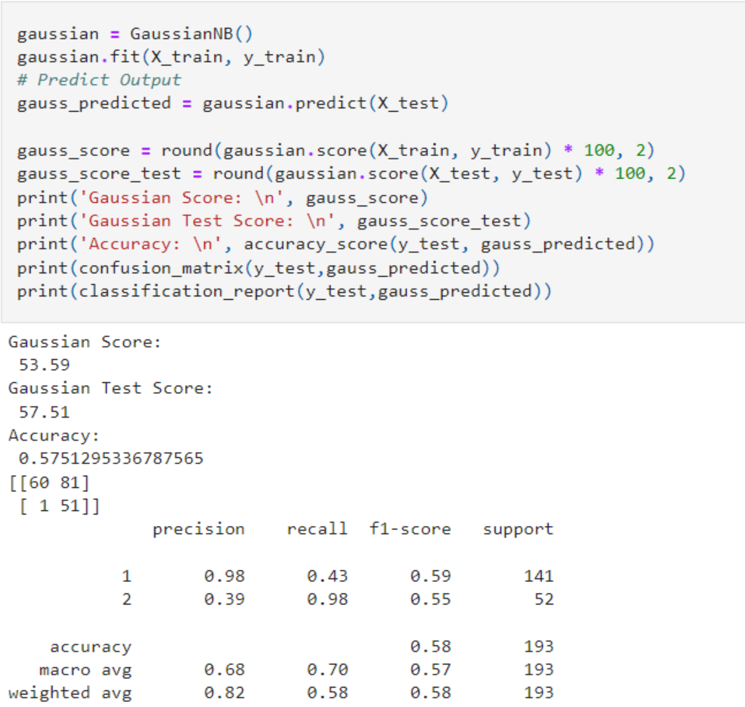
Confusion Matrix



```
sns.heatmap(confusion_matrix(y_test,log_predicted),annot=True,fmt="d")
```

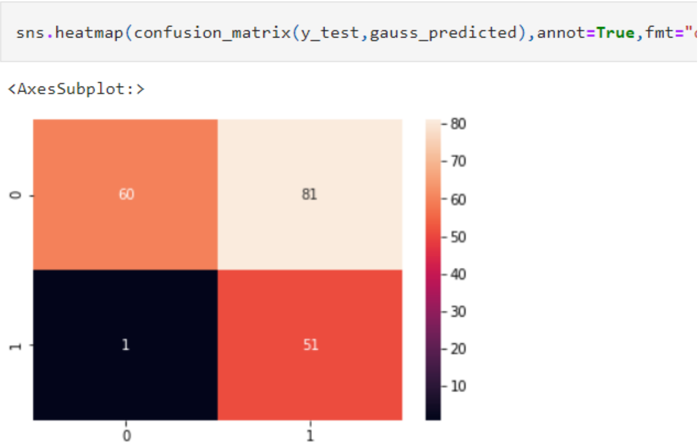
```
<AxesSubplot:>
```

2. Gaussian Naive Bayes



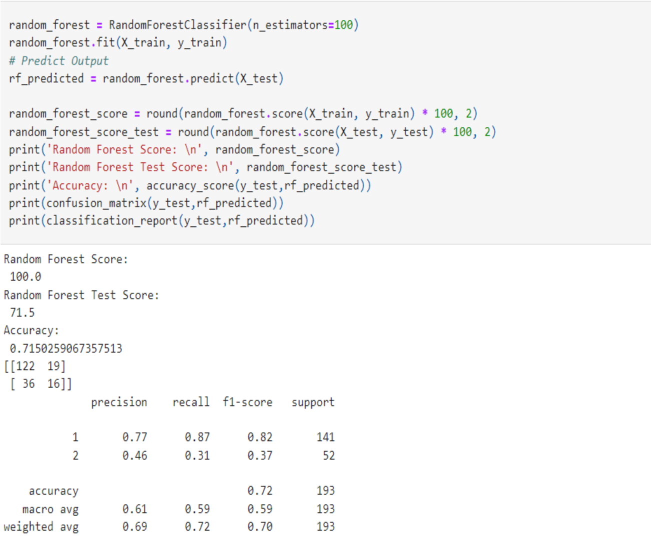
```
gaussian = GaussianNB()
gaussian.fit(X_train, y_train)
# Predict Output
gauss_predicted = gaussian.predict(X_test)

gauss_score = round(gaussian.score(X_train, y_train) * 100, 2)
gauss_score_test = round(gaussian.score(X_test, y_test) * 100, 2)
print('Gaussian Score: \n', gauss_score)
print('Gaussian Test Score: \n', gauss_score_test)
print('Accuracy: \n', accuracy_score(y_test, gauss_predicted))
print(confusion_matrix(y_test,gauss_predicted))
print(classification_report(y_test,gauss_predicted))
```

```
Gaussian Score: 
 53.59
Gaussian Test Score: 
 57.51
Accuracy: 
 0.5751295336787565
[[60 81]
 [ 1 51]]
              precision    recall  f1-score   support

           1       0.98      0.43      0.59       141
           2       0.39      0.98      0.55        52

    accuracy                           0.58       193
   macro avg       0.68      0.70      0.57       193
weighted avg       0.82      0.58      0.58       193
```



```
sns.heatmap(confusion_matrix(y_test,gauss_predicted),annot=True,fmt="
```

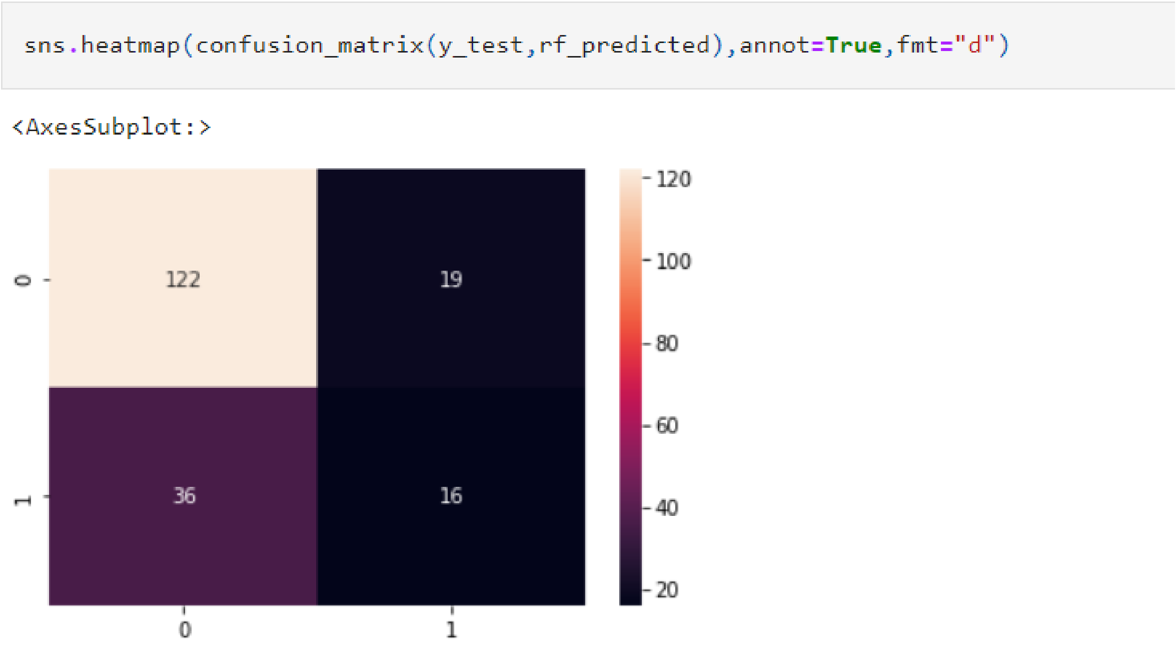
```
<AxesSubplot:>
```



```
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, y_train)
# Predict Output
rf_predicted = random_forest.predict(X_test)

random_forest_score = round(random_forest.score(X_train, y_train) * 100, 2)
random_forest_score_test = round(random_forest.score(X_test, y_test) * 100, 2)
print('Random Forest Score: \n', random_forest_score)
print('Random Forest Test Score: \n', random_forest_score_test)
print('Accuracy: \n', accuracy_score(y_test,rf_predicted))
print(confusion_matrix(y_test,rf_predicted))
print(classification_report(y_test,rf_predicted))
```

```
Random Forest Score:
 100.0
Random Forest Test Score:
 71.5
Accuracy:
 0.7150259067357513
[[122  19]
 [ 36  16]]
              precision    recall  f1-score   support

           1       0.77      0.87      0.82       141
           2       0.46      0.31      0.37        52

    accuracy                           0.72       193
   macro avg       0.61      0.59      0.59       193
weighted avg       0.69      0.72      0.70       193
```

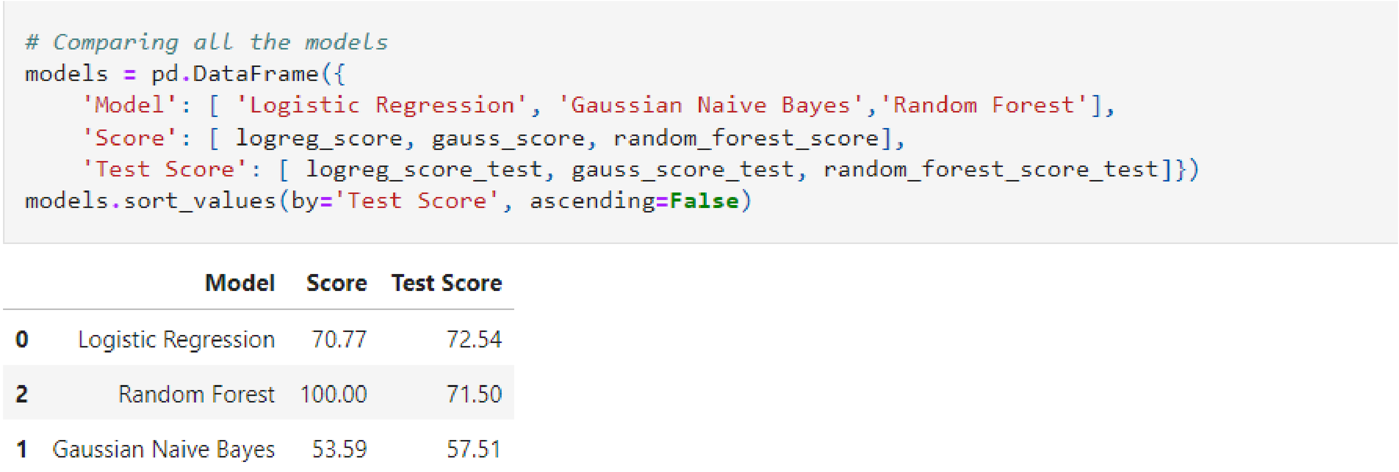
3. **Random Forest**



```
sns.heatmap(confusion_matrix(y_test,rf_predicted),annot=True,fmt="d")
```

```
<AxesSubplot:>
```

Model Evaluation



```
# Comparing all the models
models = pd.DataFrame({
    'Model': [ 'Logistic Regression', 'Gaussian Naive Bayes','Random Forest'],
    'Score': [ logreg_score, gauss_score, random_forest_score],
    'Test Score': [ logreg_score_test, gauss_score_test, random_forest_score_test]})
models.sort_values(by='Test Score', ascending=False)
```

	Model	Score	Test Score
0	Logistic Regression	70.77	72.54
2	Random Forest	100.00	71.50
1	Gaussian Naive Bayes	53.59	57.51

RESULTS AND DISCUSSION :

The project's main goal is to accurately classify patients as having liver disease or not. The Conclusion from the Models (Logistic Regression, Gaussian Naive Bayes, Random Forest) **is that the**

Logistic Regression perform the best on this dataset

CONCLUSION :

Thus we conclude a decision tree is. So, after a long journey of data visulaisation, data cleaning, data modelling etc., we have finally got our model that we can use.

REFERENCES :

[1] Namrata S Gupta, Bijendra S Agrawal, Rajkumar M. Chauhan, ìSurvey On Clustering Technique of Data Miningî, American International Journal of Research in Science, Technology, Engineering & Mathematics,ISSN:2328-3491.

[2] Malwindersingh, Meenakshibansal ,î A Survey on Various K- Means algorithms for Clusteringî, IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015.

[3] Amandeep Kaur Mann, Navneet Kaur Mann, ìReview Paper On Clustering Techniques, Global Journal Of Computer Science And Technology Software & Data Engineering, VOL. 13 ,2013.

[4] Leonid Churilov. Adyl Bagirov, Daniel Schwarta, Kate Smith, Michael Dally , Journal of management information system : 2005, Data mining with combined use of optimization techniques and self organizing maps for improving risk grouping rules : application to prostate cancer patients

[5] Anthony Danna, Oscar H. Gandy, Journal of business ethics : 2002, All that glitters is not gold : Digging Beneath the surface of data mining

[6] AC Yeo, KA Smith, RJ Willis and M Brooks, Journal of the operation research society : 2002 , A mathematical programming approach to optimize insurance premium pricing within a data minning framework.

ABOUT AUTHORS :

Dr. S. AYYASAMY obtained his bachelor’s degree in Electronics and Communication Engineering from “Maharaja Engineering College, Avinashi” under Bharathiyar University and master’s degree in Computer Science and Engineering from “PSG College of Technology, Peelamedu, Coimbatore.” under Bharathiyar University. He completed his Ph. D in Computer Science and Engineering under Anna University, Chennai. He has 21 years of teaching experience in various reputed institutions. He is currently working as Professor in School of Computer Science and Engineering, VIT University, Vellore Campus, Tamil Nadu India. He has authored more than 30 research papers in reputed international journals and well-respected international conferences. He has conducted 05 International National Conferences as a convenor. He has been working as Reviewer for reputed international journals. His Research areas of interest are Peer to Peer Networks, Overlay Networks, Soft Computing, Machine learning and Artificial Intelligence.

Ms. Abhinaya is pursuing her Int. MTech CSE (Computer Science and Engineering), 2020 – 2025, in Vellore Institute of Technology, Tamil Nadu, India

Ms. Samitha is pursuing her Int. MTech CSE (Computer Science and Engineering), 2020 – 2025, in Vellore Institute of Technology, Tamil Nadu, India

Ms. Thejashwini is pursuing her Int. MTech CSE (Computer Science and Engineering), 2020 – 2025, in Vellore Institute of Technology, Tamil Nadu, India

Soft Computing

Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computing technique

--Manuscript Draft--

Manuscript Number:	SOCO-D-23-02167
Full Title:	Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computing technique
Article Type:	S.I. : Emerging Evolutionary Computing using ML for Health Care
Keywords:	Precision; Accuracy; Regression; Liver disorder; Recall
Corresponding Author:	Samitha Reddy Vellore Institute of Technology INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Vellore Institute of Technology
First Author:	Samitha Reddy
First Author Secondary Information:	
Order of Authors:	Samitha Reddy
	Abhinaya bhimineni
	Thejashwini reddy
	S Ayyasamy
Funding Information:	
Abstract:	

Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computing technique

S. Ayyasamy
[1] *Professor*, School of Computer Science and Engineering, *Vellore Institute of Technology, Tamil Nadu, India*
ayyasamy.s@vit.ac.in

B.Abhinaya
[2] *Post Graduate student,* School of Computer Science and Engineering, *Vellore Institute of Technology, Tamil Nadu, India*
abhinaya.bhimineni2020@vitstudent.ac.in

K.Samitha
[3] *Post Graduate of* School of Computer Science and Engineering, *Vellore Institute of Technology, TamilNadu, India*
kottam.samitha2020@vitstudent.ac.in

R.Thejashwini
[4] *Post Graduate of* School of Computer Science and Engineering, *Vellore Institute of Technology, TamilNadu, India*
thejashwini.r2020@vitstudent.ac.in

Abstract - This study uses a 'Logistic Regression', 'Gaussian Naive Bayes', 'Random Forest' method to try and achieve effective early diagnosis of liver illness. Using the UCI repository, we gathered 583 records pertaining to the Indian Liver Patient Dataset. 70% of the ILPD dataset is used for training, and 30% is used for testing. statistics of Indian liver patients Age, gender, total bilirubin, direct bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT, and alpha's are the 10 variables in this equation. As well as determining the overall accuracy, we will determine if the person has liver disease.

Keywords:

Precision, Accuracy, Regression, Liver disorder, Recall

Introduction:

The liver controls a number of potentially harmful bodily processes, and if it develops a disease or is destroyed, the body may suffer serious harm as a result of the lack of those processes. Hepatic disease is another name for liver disease. The broad phrase "liver disease" refers to all possible issues that could prevent the liver from carrying out its intended duties. Typically, three quarters or more of the liver's tissue must be damaged before liver function starts to decline.

This paper describes the approach, one of the most used supervised classification methods. The use of classification systems in various automatic medical diagnostics is very common. While the liver will continue to operate correctly even when it is partially damaged, problems with liver patients are difficult to identify at an early stage. The likelihood that a patient will survive will rise with an early diagnosis of liver issues. Enzyme levels in the blood can be analysed to diagnose liver disease. Age, gender, total bilirubin, direct bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT, and Alpo's are the 10 variables in the Indian liver patient dataset T.

Nowadays, medical professionals frequently employ artificial intelligence to detect a variety of illnesses that are brought on by the dysfunction of particular organs.

Literature survey :

Authors	Methodology or Techniques used	Advantages	Issues	Metrics used
1)Jeddah	Genetic algorithms, Computer assisted diagnosis	improve the efficiency and effectiveness of the admission process	The improved method avoids computing the distance of each data object to the cluster centers repeatly, saving the running time	Supervised learning technique
2)Himani Sharma	ANN, Back propagation diagnosis, Feed Forward Neural Network	Accuracy was increased by 1%.	This population also appears to be predisposed to developing this disease earlier, compared to the Western population	Decision tree
3)Mr. Brijain R Patel, Mr. Kushik K	Decision tree, Back propagation Neural Network	Accuracy was increased by 2%	focus on the various algorithms	Classification and prediction are the techniques used to make out important data classes and predict probable

4)Huang Ming	Data mining classification, Neural Networks, Parallelism	simplifies the information entropy solution of ID3 algorithm	Alcoholic liver disease (ALD) is one of the main causes of chronic liver disease worldwide	It accounts for up to 48% of cirrhosis-associated deaths
5) Niu Wenying	Back propogation networks, Genetic algorithm	accracy was increased by 5%	It has been reported that haemodialysis increases the possibility of blood borne viral infection but the prevalence is variable from haemodialysis from centre to centre and also from region to region and country to country, and high-cost haemodialysis centre vs low-cost haemodialysis centre.	In most of the study, HBV infection among hemodialysis patient was between 4 and 11%
6) Vaidya, M.HChaudri	Artificial neural Networks, Fuzzy logic, Fuzzy Neural Network, Classification,	Early diagnosis is of considerable amount of significance in treating the disease. Diagnosis is of the physician skills conducting based on their knowledge's and experience yet an error might occurrence is here	It cannot be a lot of possible errors in this diagnosis due to the number of enzymes to be many as well as the effects of different taken alcohol rates to be very from one patient to the other.	The Liver Disorders includes 345 specimens consisting of six fields and two classes. Each sample is taken from an single man. Two hundred of these samples are of one class with remaining 145 are possessed by to the other.
7) Vijayarani.s Dhayanand.s	Artificial Neural Network (ANN) classification algorithm. LS-	This dataset contains Liver Function Test details (LFT).	Utilized PC and LSSVM doesn't give the expected results	Diabetes Dataset Indian Liver Patient Dataset (ILPD). Dataset contains Liver Function Test details (LFT).

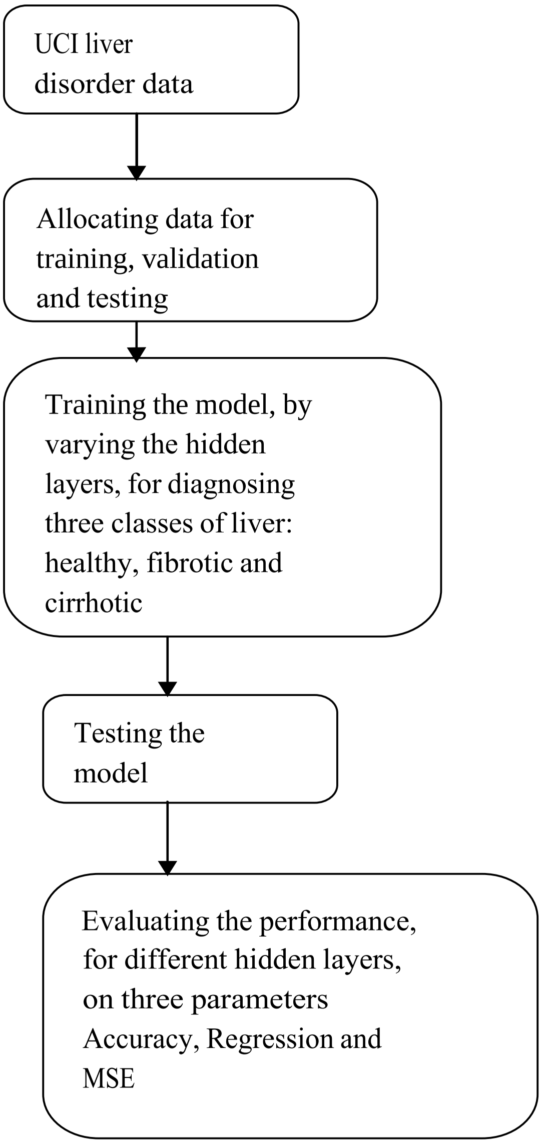
	SVM algorithm	Karthik et.al were applied a soft computing technique for intelligent diagnosis of liver disease. They have implemented classification and its type detection in phases.		
8) Lin R.H	Random forest algorithm, classification, computational intelligence,	It is shown that feature selection has a great significance as the process of selecting a subset of relevant features for use in model construction. By using feature selection on ILPD before a classification algorithm can be applied, performance of classification algorithm increases.	Problems with liver patients are not easily discovered in an early stage as it will be functioning normally even when it is partially damaged [2]. An early diagnosis of liver problems will increase patient's survival rate.	Classifying Banking Dataset, Indian Liver Patient Dataset(ILDP)
9) Jankisharan Pahareeya Rajan Vohra Jagdish Makhijani Sanjay Patsariya	Multilayer Feed Forward Neural Network, Random Forest, Multiple Linear Regression (MLR), Support Vector Machine (SVM) and Genetic programming (GP).	The results indicates that there exists more significant difference in the groups with all the possible attribute combinations except analysis on SGPT between non liver patients of UCI and INDIA data sets	the accuracy of these models is not satisfactory so there is always a scope for new classifactory models.	ILPD data set and UCI data set
10) Kalyan	Discriminative	To serve the	Identification of	It was followed by splitting of

Nagaraj and Amulyashree Sridhar	learning, Artificial Neural Network, Bagging, Boosting, Naïve Bayes, Kernel-based classifiers, Nearest Neighbour algorithm, Decision Trees, Random Forest,	medicinal community for prediction of liver disease among patients, a graphical user interface (GUI) has been developed using R. The GUI is deployed as a package in local repository of R platform for users to perform prediction.	liver infection at preliminary stage is important but combat the frequency and severity deaths of patients in India are higher. The patients must be screened based on initial symptoms for development of personalized therapy.	the dataset into training (70% of the dataset) and test (30%) sets. Training set comprised of 389 instances and test set included the remaining 194 instances.

PROPOSED WORK :

ARCHITECTURE :

Fig-1


UCI liver disorder data
Allocating data for training, validation and testing
Training the model, by varying the hidden layers, for diagnosing three classes of liver: healthy, fibrotic and cirrhotic
Testing the model
Evaluating the performance, for different hidden layers, on three parameters Accuracy, Regression and MSE

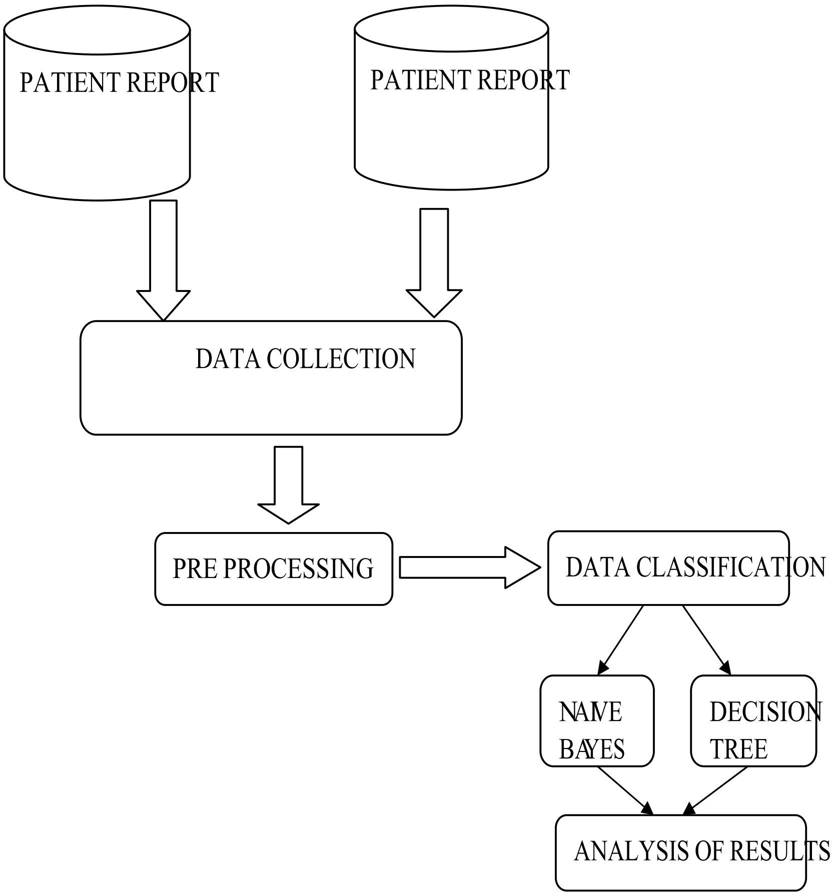


Fig-2

DECISION TREE :

Using Euclidean distance similarity, divide the training cases into k clusters. We construct decision trees using the C4.5 decision tree technique on each cluster, which represents a density region of typical or anomalous cases.

PRE PROCESSING:

Characteristics must range from 0 to 1. The action is known as normalisation. Each sample of a particular property is normalised by dividing it by its greatest value.

Gaussian Naive Bayes

When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Sometimes assume variance

- is independent of Y (i.e., σi)
- or independent of Xi (i.e., σk)
- or both (i.e., σ)

PATIENT REPORT:

The patient report is very important.Since the patient has access to all information regarding their diagnosis, medical history, prescriptions, and appointment times. It must not be mixed up with any other patient

Evaluation Metrics Used -

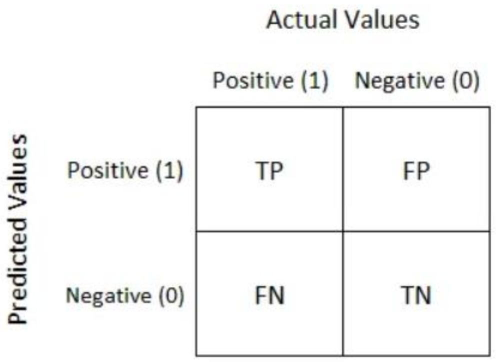
Since this is binary classification problem, we use the following metrics:

- Confusion matrix - For getting a better clarity of the no of correct/incorrect predictions by the model.

In order for the classifier to work at its best, the attribute values must be converted into homogenous, well-behaved values that generate numerical stability. As a result, the values of the   patients. They must very carefully safeguard the patient data. It shouldn’t be in a risky situation. Data gathering is a crucial procedure. Data shouldn’t be mixed up with patient information

DATA COLLECTION: Here Data is collected and we perform the required methods.

The Indian Liver Patient Dataset collects patient data, which is then stored in several databases. They collect the data, analyse it, and then communicate the findings to the information. Moreover, it never exchanges by error. The patient report must always be given to the appropriate patients.



Predicted Values		Actual Values: Positive (1)	Actual Values: Negative (0)
	Positive (1)	TP	FP
	Negative (0)	FN	TN

Confusion Metrics

From our confusion matrix, we can calculate five

different metrics measuring the validity of our model.

1. Accuracy (all **correct** / all)

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

2. Misclassification (all **incorrect** / all) = FP + FN / TP + TN + FP + FN

3. Precision (**true** positives / **predicted** positives) =

$$Precision = \frac{TP}{TP + FP}$$

Sensitivity aka Recall (**true** positives / all **actual** positives) =

$$Recall = \frac{TP}{TP + FN}$$

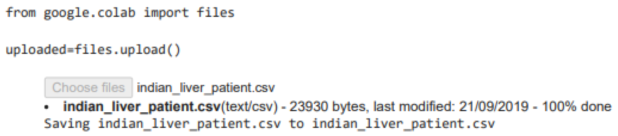
Specificity (**true** negatives / all **actual** negatives) =TN / TN + FP

4) F1 score

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

EXPERIMENTS AND RESULTS:

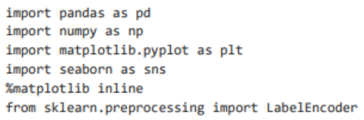
Analysis and prediction of Indian liver patient



```
from google.colab import files

uploaded=files.upload()
```

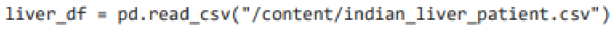
- **indian_liver_patient.csv**(text/csv) - 23930 bytes, last modified: 21/09/2019 - 100% done

```
Saving indian_liver_patient.csv to indian_liver_patient.csv
```

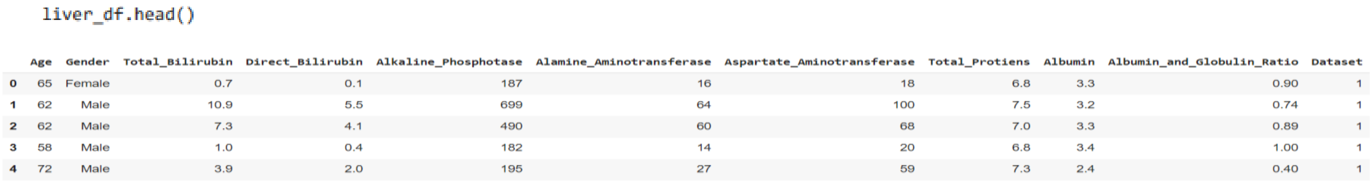


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import LabelEncoder
```

Data Analysis:

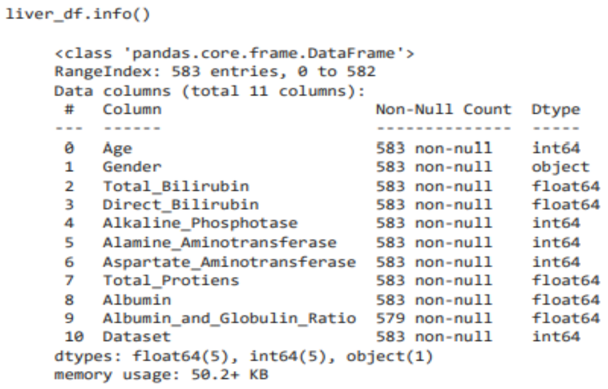


```
liver_df = pd.read_csv("/content/indian_liver_patient.csv")
```



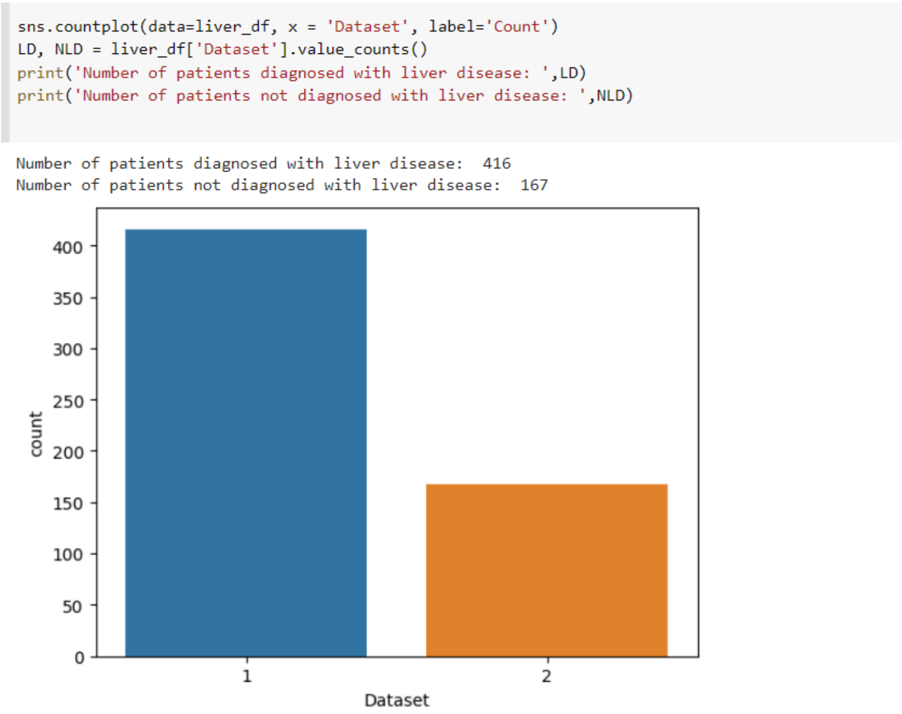
```
liver_df.head()
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
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



```
liver_df.info()
```

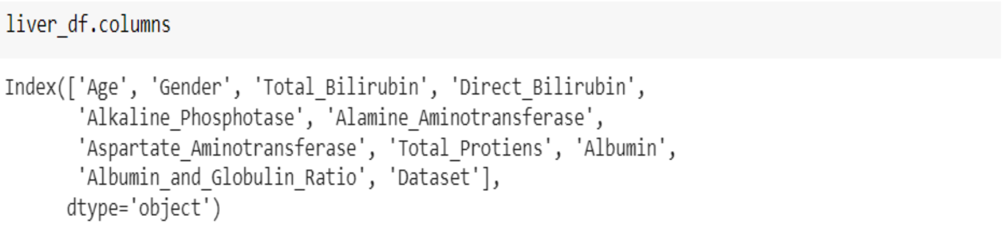
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         583 non-null    int64
 1   Gender                      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   Alamine_Aminotransferase    583 non-null    int64
 6   Aspartate_Aminotransferase  583 non-null    int64
 7   Total_Protiens              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: 50.2+ KB
```



```
sns.countplot(data=liver_df, x = 'Dataset', label='Count')
LD, NLD = liver_df['Dataset'].value_counts()
print('Number of patients diagnosed with liver disease: ',LD)
print('Number of patients not diagnosed with liver disease: ',NLD)
```

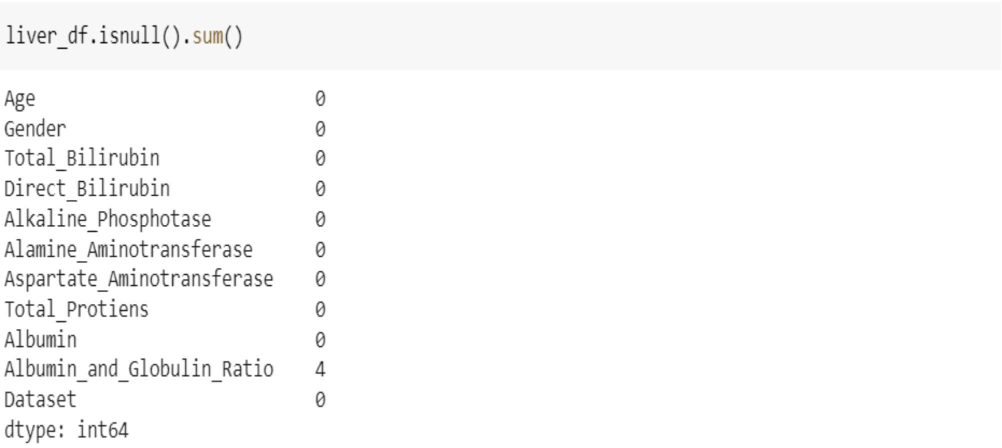
```
Number of patients diagnosed with liver disease:  416
Number of patients not diagnosed with liver disease:  167
```

Fig-3



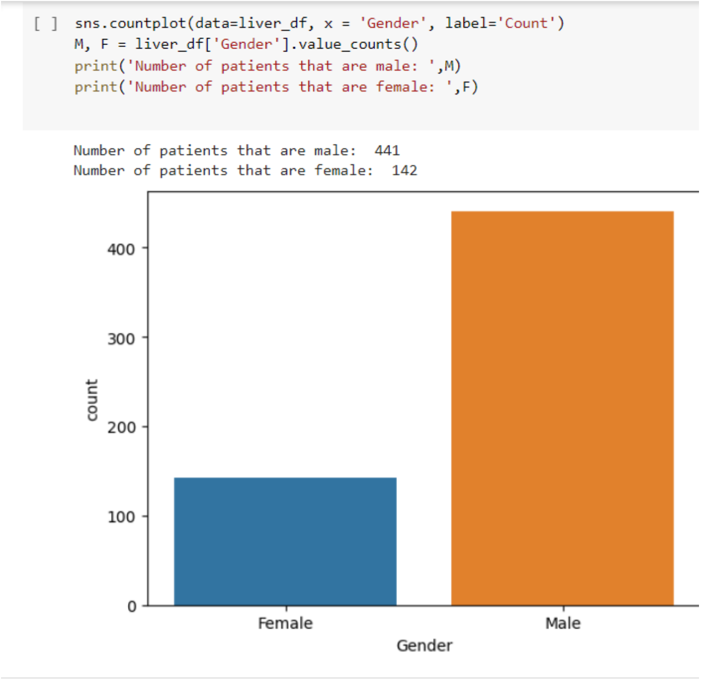
```
liver_df.columns
```

```
Index(['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
       'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
       'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
       'Albumin_and_Globulin_Ratio', 'Dataset'],
      dtype='object')
```



```
liver_df.isnull().sum()
```

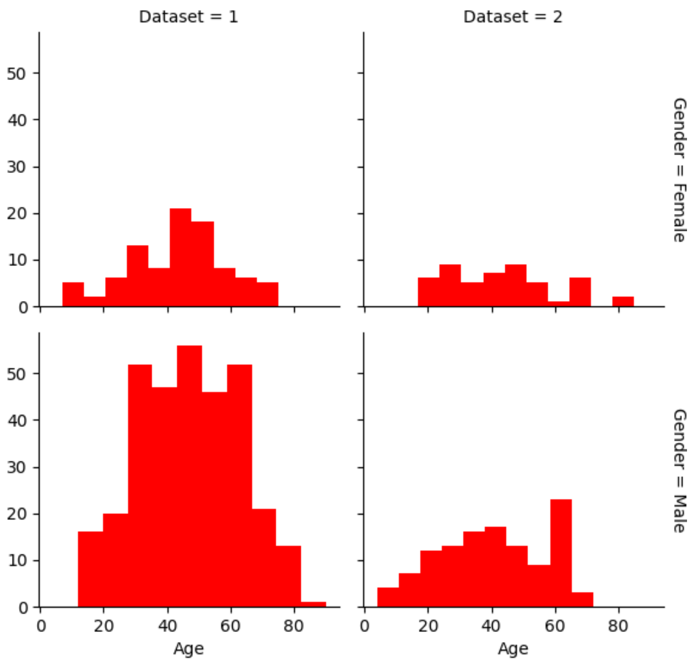
```
Age                           0
Gender                        0
Total_Bilirubin               0
Direct_Bilirubin              0
Alkaline_Phosphotase          0
Alamine_Aminotransferase      0
Aspartate_Aminotransferase    0
Total_Protiens                0
Albumin                       0
Albumin_and_Globulin_Ratio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

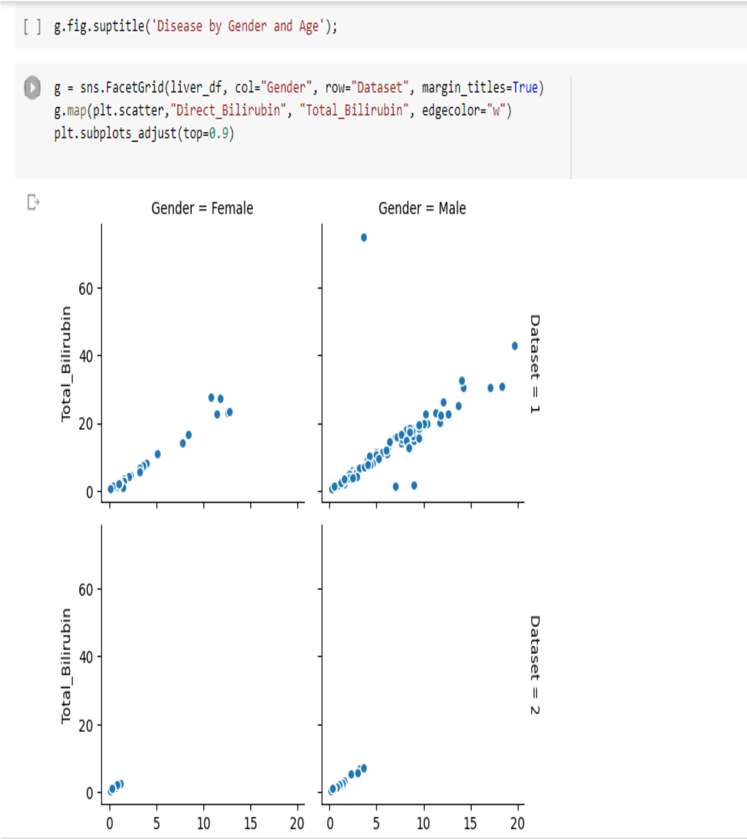


```
[ ] sns.countplot(data=liver_df, x = 'Gender', label='Count')
    M, F = liver_df['Gender'].value_counts()
    print('Number of patients that are male: ',M)
    print('Number of patients that are female: ',F)
```

```
Number of patients that are male:  441
Number of patients that are female:  142
```

Fig-4

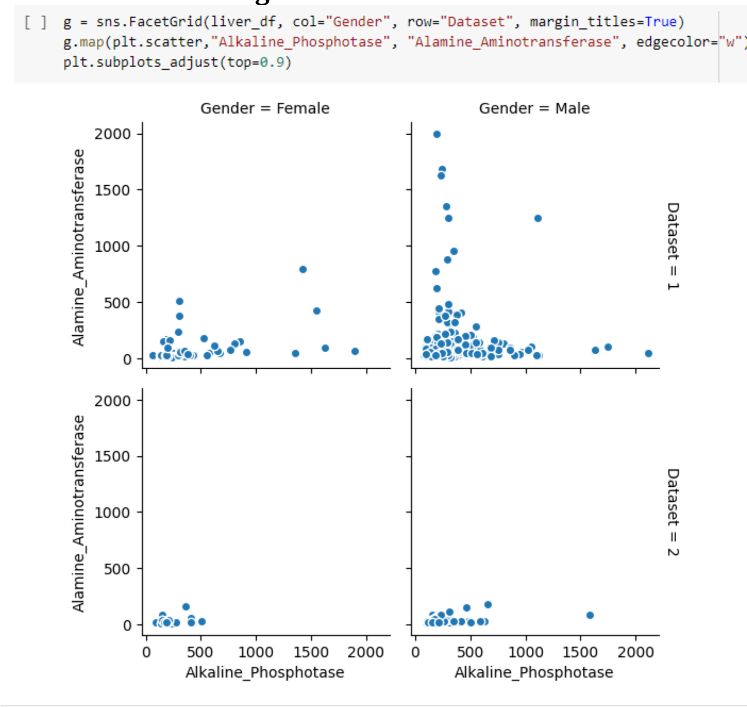




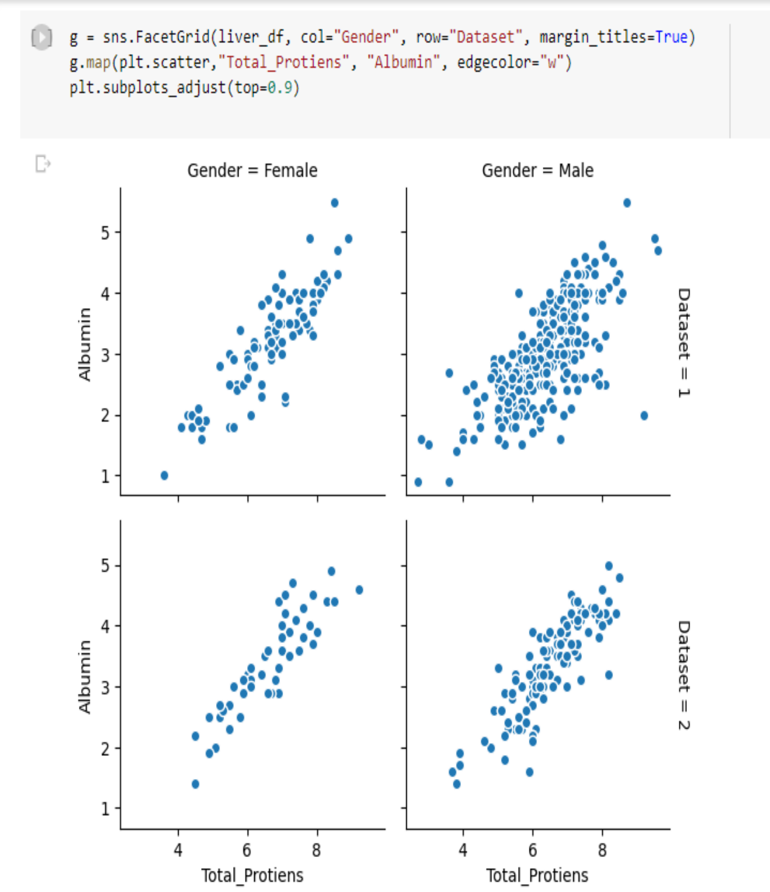
```
[ ] g.fig.suptitle('Disease by Gender and Age');
```

```
g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
g.map(plt.scatter,"Direct_Bilirubin", "Total_Bilirubin", edgecolor="w")
plt.subplots_adjust(top=0.9)
```

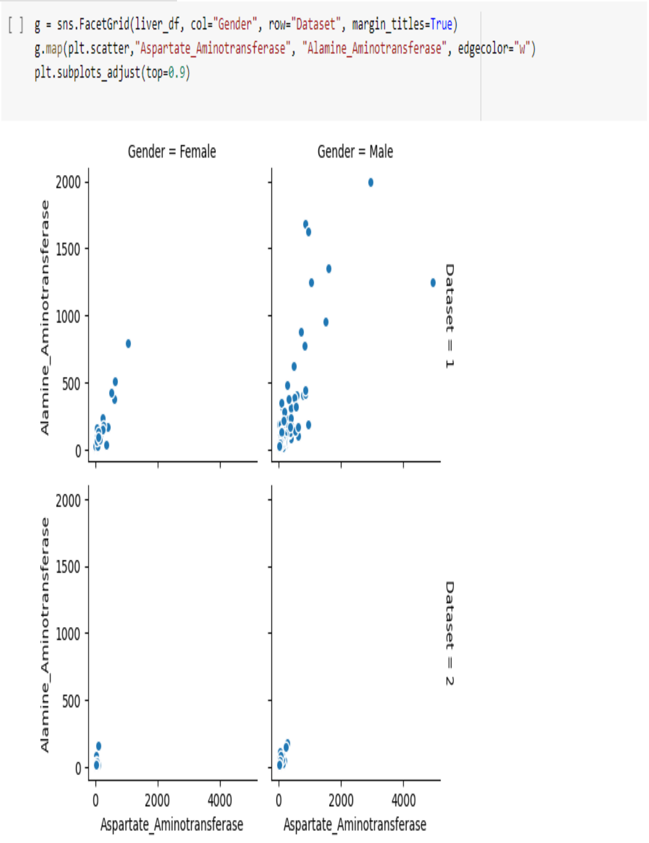
Fig-5



```
[ ] g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
    g.map(plt.scatter,"Alkaline_Phosphotase", "Alamine_Aminotransferase", edgecolor="w"
    plt.subplots_adjust(top=0.9)
```

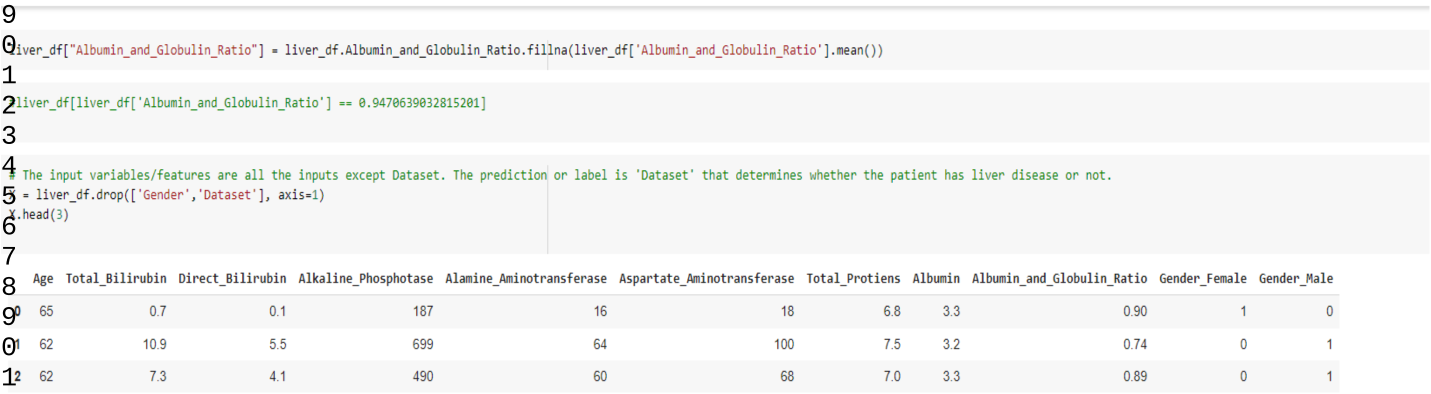


```
g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
g.map(plt.scatter,"Total_Protiens", "Albumin", edgecolor="w")
plt.subplots_adjust(top=0.9)
```



```
[ ] g = sns.FacetGrid(liver_df, col="Gender", row="Dataset", margin_titles=True)
    g.map(plt.scatter,"Aspartate_Aminotransferase", "Alamine_Aminotransferase", edgecolor="w")
    plt.subplots_adjust(top=0.9)
```

Fig-6 **Fig-7**

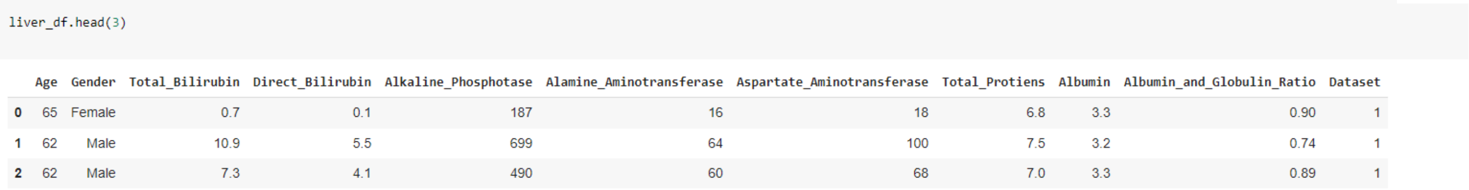


```
liver_df["Albumin_and_Globulin_Ratio"] = liver_df.Albumin_and_Globulin_Ratio.fillna(liver_df['Albumin_and_Globulin_Ratio'].mean())
```

```
liver_df[liver_df['Albumin_and_Globulin_Ratio'] == 0.9470639032815201]
```

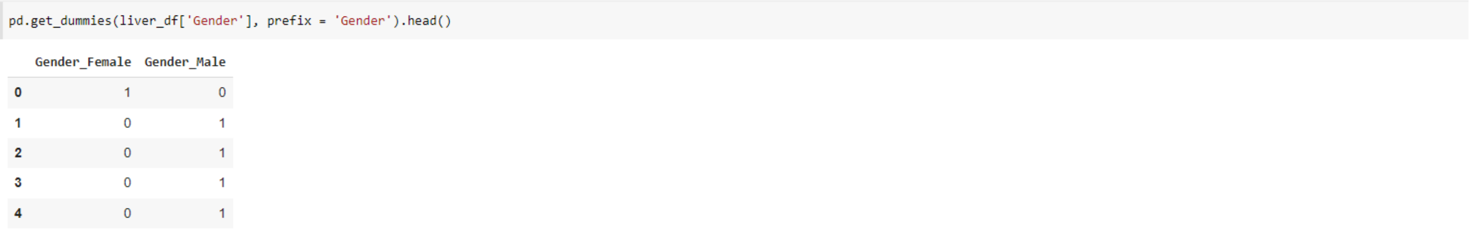
```
# The input variables/features are all the inputs except Dataset. The prediction or label is 'Dataset' that determines whether the patient has liver disease or not.
X = liver_df.drop(['Gender','Dataset'], axis=1)
X.head(3)
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Gender_Female	Gender_Male
0	65	0.7	0.1	187	16	18	6.8	3.3	0.90	1	0
1	62	10.9	5.5	699	64	100	7.5	3.2	0.74	0	1
2	62	7.3	4.1	490	60	68	7.0	3.3	0.89	0	1



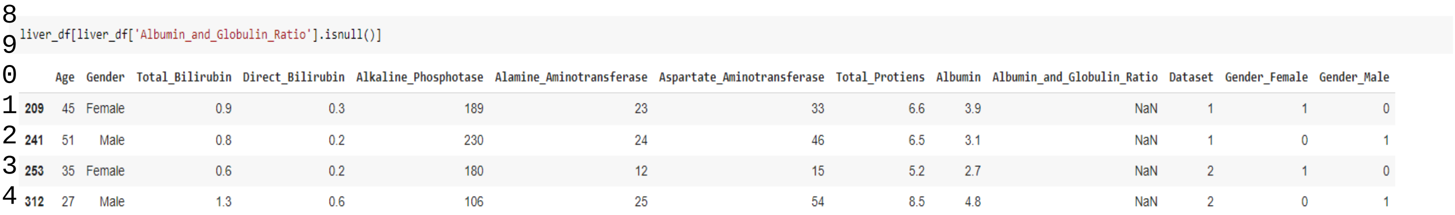
```
liver_df.head(3)
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
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



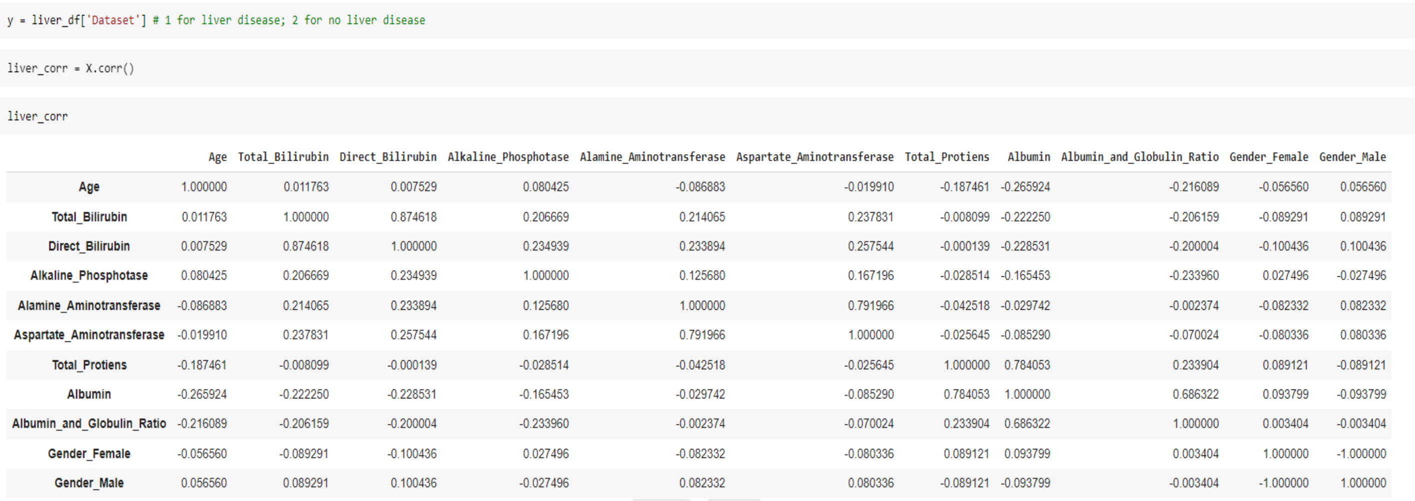
```
pd.get_dummies(liver_df['Gender'], prefix = 'Gender').head()
```

	Gender_Female	Gender_Male
0	1	0
1	0	1
2	0	1
3	0	1
4	0	1



```
liver_df[liver_df['Albumin_and_Globulin_Ratio'].isnull()]
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset	Gender_Female	Gender_Male
209	45	Female	0.9	0.3	189	23	33	6.6	3.9	NaN	1	1	0
241	51	Male	0.8	0.2	230	24	46	6.5	3.1	NaN	1	0	1
253	35	Female	0.6	0.2	180	12	15	5.2	2.7	NaN	2	1	0
312	27	Male	1.3	0.6	106	25	54	8.5	4.8	NaN	2	0	1

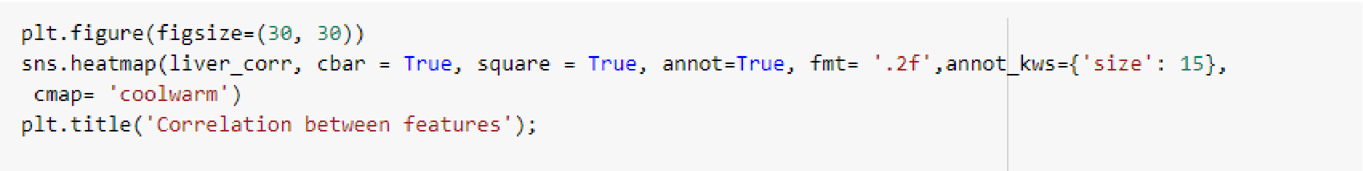


```
y = liver_df['Dataset'] # 1 for liver disease; 2 for no liver disease
```

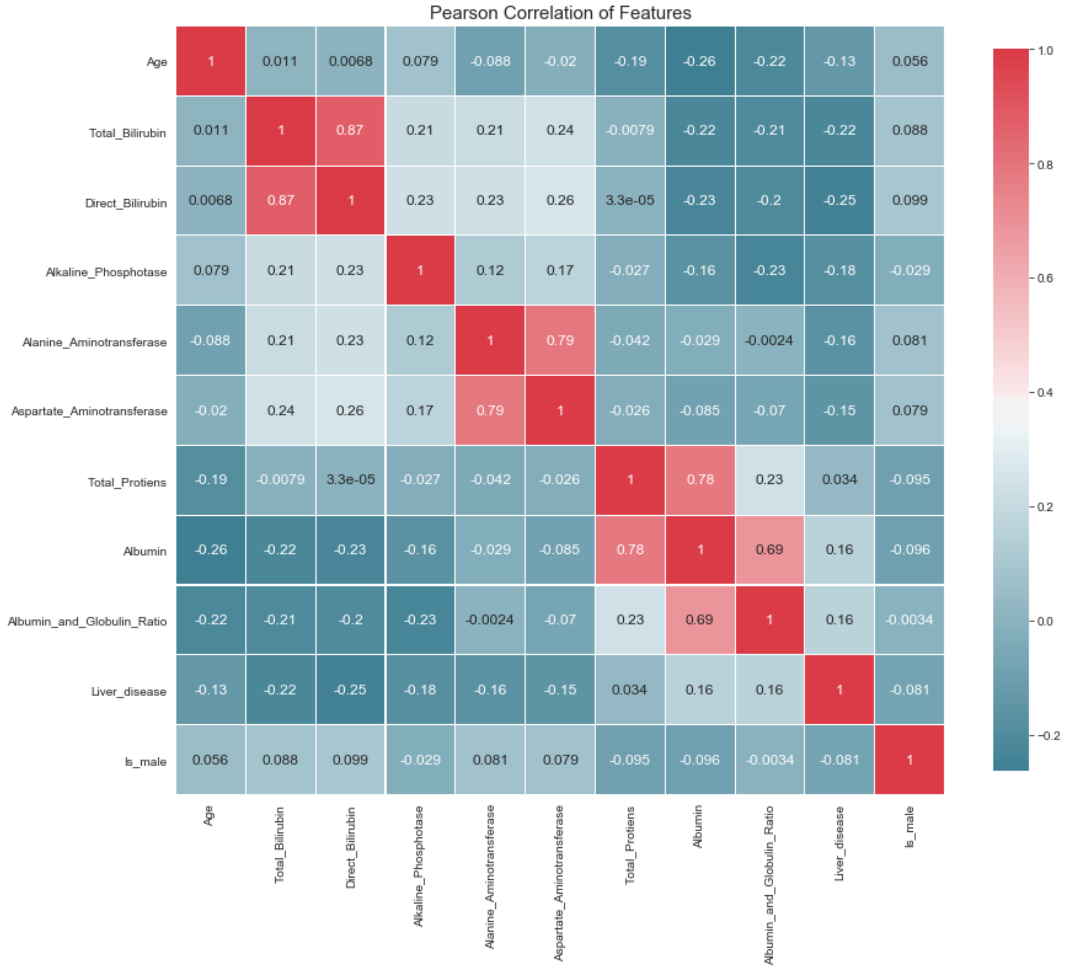
```
liver_corr = X.corr()
```

```
liver_corr
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Gender_Female	Gender_Male
Age	1.000000	0.011763	0.007529	0.080425	-0.086883	-0.019910	-0.187461	-0.265924	-0.216089	-0.056560	0.056560
Total_Bilirubin	0.011763	1.000000	0.874618	0.206669	0.214065	0.237831	-0.008099	-0.222250	-0.206159	-0.089291	0.089291
Direct_Bilirubin	0.007529	0.874618	1.000000	0.234939	0.233894	0.257544	-0.000139	-0.228531	-0.200004	-0.100436	0.100436
Alkaline_Phosphotase	0.080425	0.206669	0.234939	1.000000	0.125680	0.167196	-0.028514	-0.165453	-0.233960	0.027496	-0.027496
Alamine_Aminotransferase	-0.086883	0.214065	0.233894	0.125680	1.000000	0.791966	-0.042518	-0.029742	-0.002374	-0.082332	0.082332
Aspartate_Aminotransferase	-0.019910	0.237831	0.257544	0.167196	0.791966	1.000000	-0.025645	-0.085290	-0.070024	-0.080336	0.080336
Total_Protiens	-0.187461	-0.008099	-0.000139	-0.028514	-0.042518	-0.025645	1.000000	0.784053	0.233904	0.089121	-0.089121
Albumin	-0.265924	-0.222250	-0.228531	-0.165453	-0.029742	-0.085290	0.784053	1.000000	0.686322	0.093799	-0.093799
Albumin_and_Globulin_Ratio	-0.216089	-0.206159	-0.200004	-0.233960	-0.002374	-0.070024	0.233904	0.686322	1.000000	0.003404	-0.003404
Gender_Female	-0.056560	-0.089291	-0.100436	0.027496	-0.082332	-0.080336	0.089121	0.093799	0.003404	1.000000	-1.000000
Gender_Male	0.056560	0.089291	0.100436	-0.027496	0.082332	0.080336	-0.089121	-0.093799	-0.003404	-1.000000	1.000000



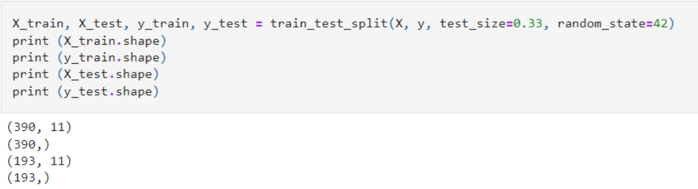
```
plt.figure(figsize=(30, 30))
sns.heatmap(liver_corr, cbar = True, square = True, annot=True, fmt= '.2f',annot_kws={'size': 15},
 cmap= 'coolwarm')
plt.title('Correlation between features');
```



	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alanine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Liver_disease	Is_male
Age	1	0.011	0.0068	0.079	-0.088	-0.02	-0.19	-0.26	-0.22	-0.13	0.056
Total_Bilirubin	0.011	1	0.87	0.21	0.21	0.24	-0.0079	-0.22	-0.21	-0.22	0.088
Direct_Bilirubin	0.0068	0.87	1	0.23	0.23	0.26	3.3e-05	-0.23	-0.2	-0.25	0.099
Alkaline_Phosphotase	0.079	0.21	0.23	1	0.12	0.17	-0.027	-0.16	-0.23	-0.18	-0.029
Alanine_Aminotransferase	-0.088	0.21	0.23	0.12	1	0.79	-0.042	-0.029	-0.0024	-0.16	0.081
Aspartate_Aminotransferase	-0.02	0.24	0.26	0.17	0.79	1	-0.026	-0.085	-0.07	-0.15	0.079
Total_Protiens	-0.19	-0.0079	3.3e-05	-0.027	-0.042	-0.026	1	0.78	0.23	0.034	-0.095
Albumin	-0.26	-0.22	-0.23	-0.16	-0.029	-0.085	0.78	1	0.69	0.16	-0.096
Albumin_and_Globulin_Ratio	-0.22	-0.21	-0.2	-0.23	-0.0024	-0.07	0.23	0.69	1	0.16	-0.0034
Liver_disease	-0.13	-0.22	-0.25	-0.18	-0.16	-0.15	0.034	0.16	0.16	1	-0.081
Is_male	0.056	0.088	0.099	-0.029	0.081	0.079	-0.095	-0.096	-0.0034	-0.081	1

Fig-8

Splitting the data into Train and Test

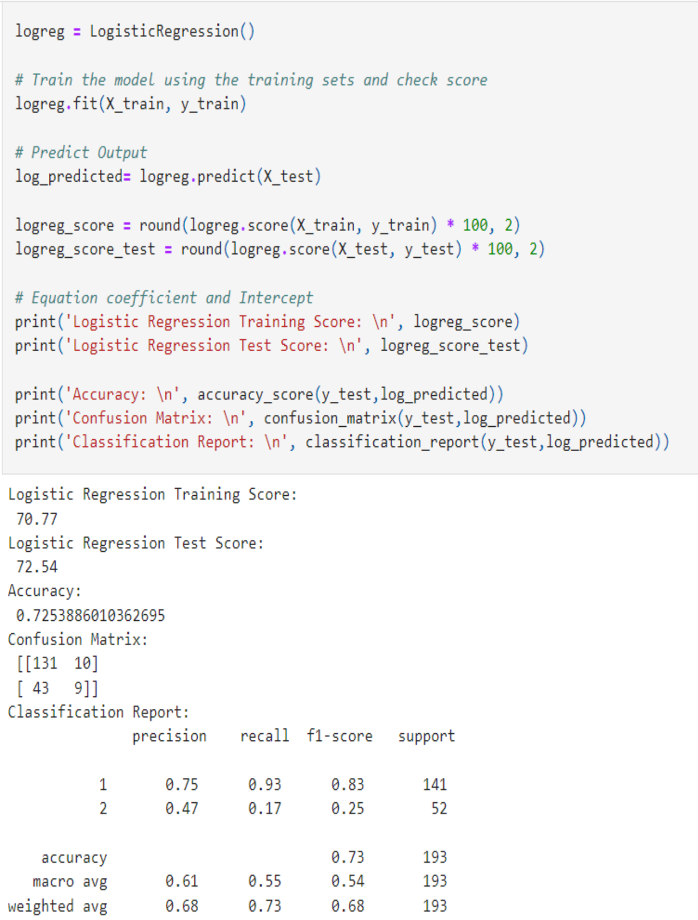


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
print (X_train.shape)
print (y_train.shape)
print (X_test.shape)
print (y_test.shape)
```

```
(390, 11)
(390,)
(193, 11)
(193,)
```

Model Building

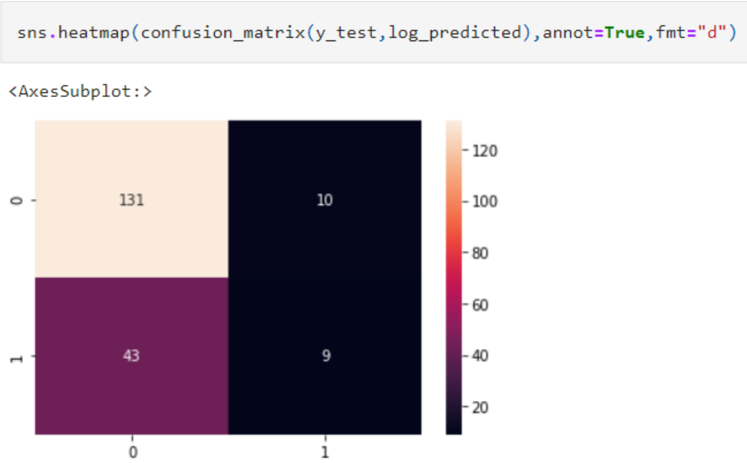
1. Logistic Regression



```
logreg = LogisticRegression()

# Train the model using the training sets and check score
logreg.fit(X_train, y_train)

# Predict Output
log_predicted= logreg.predict(X_test)

logreg_score = round(logreg.score(X_train, y_train) * 100, 2)
logreg_score_test = round(logreg.score(X_test, y_test) * 100, 2)

# Equation coefficient and Intercept
print('Logistic Regression Training Score: \n', logreg_score)
print('Logistic Regression Test Score: \n', logreg_score_test)

print('Accuracy: \n', accuracy_score(y_test,log_predicted))
print('Confusion Matrix: \n', confusion_matrix(y_test,log_predicted))
print('Classification Report: \n', classification_report(y_test,log_predicted))
```

```
Logistic Regression Training Score: 
 70.77
Logistic Regression Test Score: 
 72.54
Accuracy: 
 0.7253886010362695
Confusion Matrix: 
 [[131  10]
 [ 43   9]]
Classification Report: 
               precision    recall  f1-score   support

           1       0.75      0.93      0.83       141
           2       0.47      0.17      0.25        52

    accuracy                           0.73       193
   macro avg       0.61      0.55      0.54       193
weighted avg       0.68      0.73      0.68       193
```

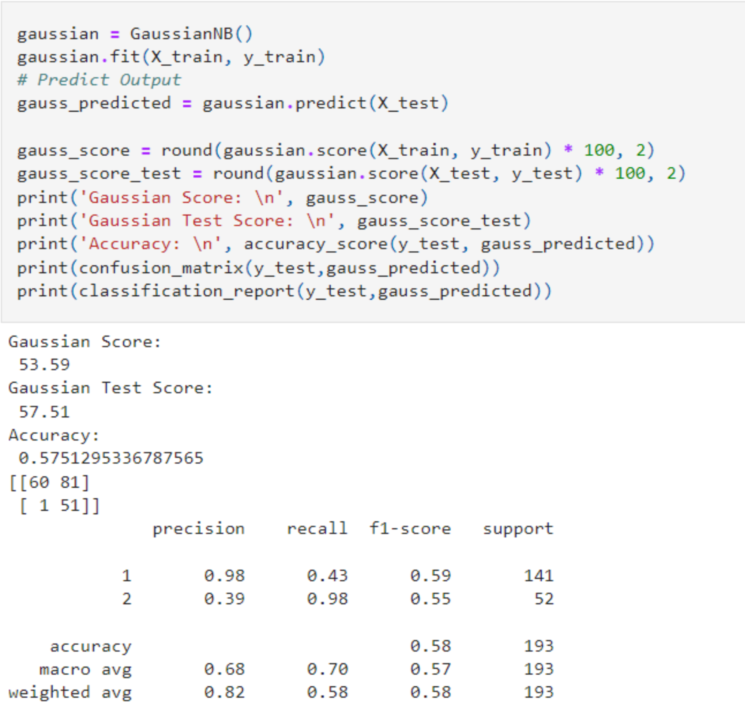
Confusion Matrix



```
sns.heatmap(confusion_matrix(y_test,log_predicted),annot=True,fmt="d")
```

```
<AxesSubplot:>
```

2. Gaussian Naive Bayes

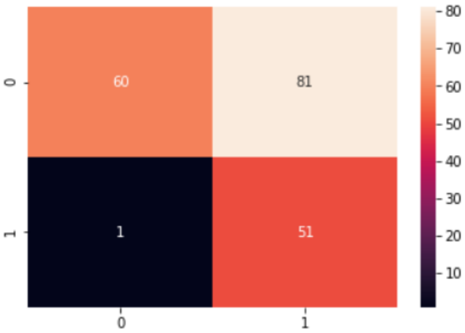


```
gaussian = GaussianNB()
gaussian.fit(X_train, y_train)
# Predict Output
gauss_predicted = gaussian.predict(X_test)

gauss_score = round(gaussian.score(X_train, y_train) * 100, 2)
gauss_score_test = round(gaussian.score(X_test, y_test) * 100, 2)
print('Gaussian Score: \n', gauss_score)
print('Gaussian Test Score: \n', gauss_score_test)
print('Accuracy: \n', accuracy_score(y_test, gauss_predicted))
print(confusion_matrix(y_test,gauss_predicted))
print(classification_report(y_test,gauss_predicted))
```

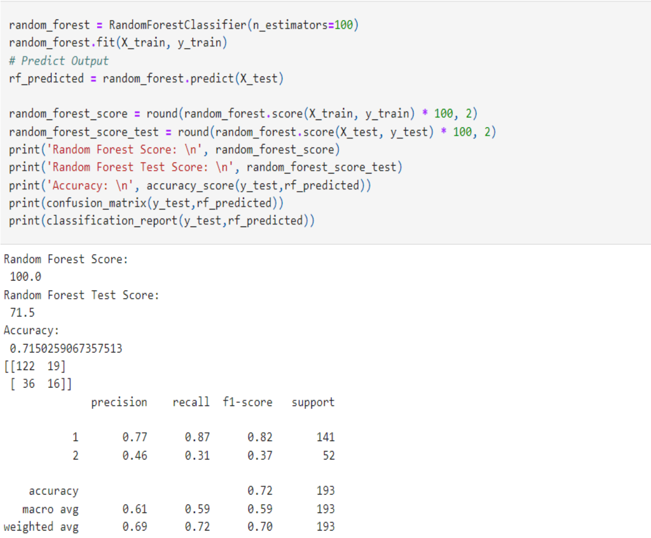
```
Gaussian Score: 
 53.59
Gaussian Test Score: 
 57.51
Accuracy: 
 0.5751295336787565
[[60 81]
 [ 1 51]]
              precision    recall  f1-score   support

           1       0.98      0.43      0.59       141
           2       0.39      0.98      0.55        52

    accuracy                           0.58       193
   macro avg       0.68      0.70      0.57       193
weighted avg       0.82      0.58      0.58       193
```

```
sns.heatmap(confusion_matrix(y_test,gauss_predicted),annot=True,fmt="(
```

```
<AxesSubplot:>
```

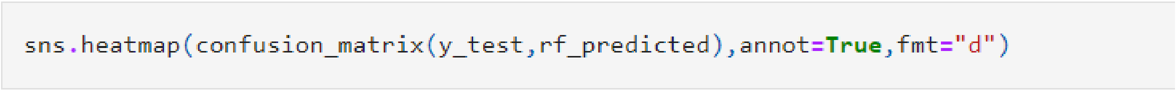


3. Random Forest



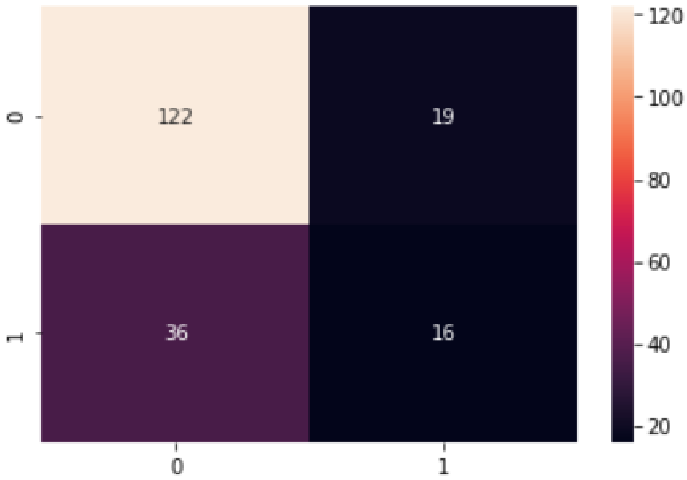
```
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, y_train)
# Predict Output
rf_predicted = random_forest.predict(X_test)

random_forest_score = round(random_forest.score(X_train, y_train) * 100, 2)
random_forest_score_test = round(random_forest.score(X_test, y_test) * 100, 2)
print('Random Forest Score: \n', random_forest_score)
print('Random Forest Test Score: \n', random_forest_score_test)
print('Accuracy: \n', accuracy_score(y_test,rf_predicted))
print(confusion_matrix(y_test,rf_predicted))
print(classification_report(y_test,rf_predicted))
```

```
Random Forest Score:
 100.0
Random Forest Test Score:
 71.5
Accuracy:
 0.7150259067357513
[[122  19]
 [ 36  16]]
              precision    recall  f1-score   support

           1       0.77      0.87      0.82       141
           2       0.46      0.31      0.37        52

    accuracy                           0.72       193
   macro avg       0.61      0.59      0.59       193
weighted avg       0.69      0.72      0.70       193
```

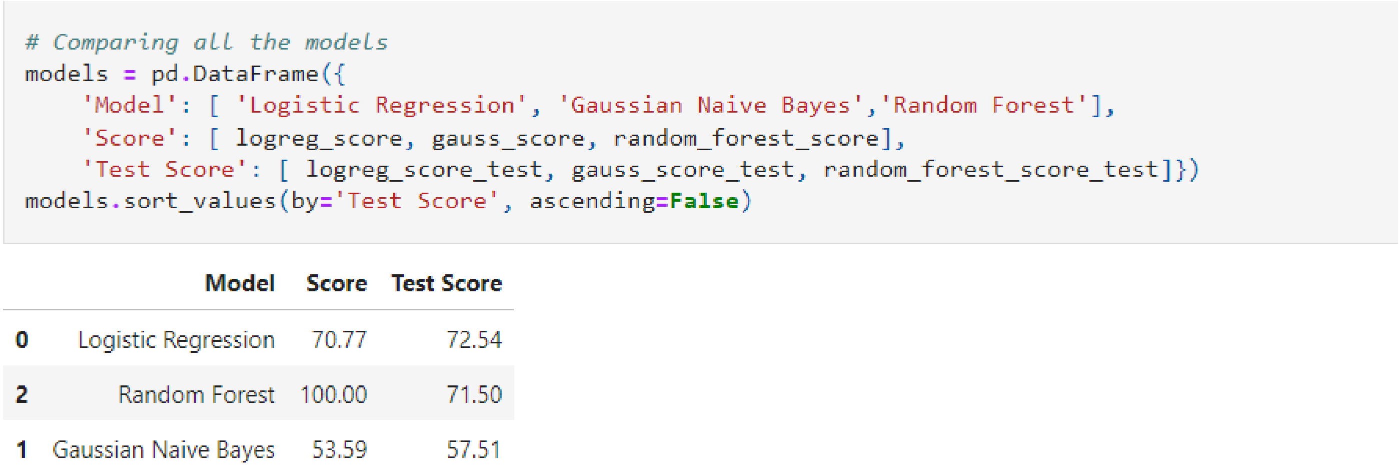


```
sns.heatmap(confusion_matrix(y_test,rf_predicted),annot=True,fmt="d")
```

```
<AxesSubplot:>
```



Model Evaluation



```
# Comparing all the models
models = pd.DataFrame({
    'Model': [ 'Logistic Regression', 'Gaussian Naive Bayes','Random Forest'],
    'Score': [ logreg_score, gauss_score, random_forest_score],
    'Test Score': [ logreg_score_test, gauss_score_test, random_forest_score_test]})
models.sort_values(by='Test Score', ascending=False)
```

	Model	Score	Test Score
0	Logistic Regression	70.77	72.54
2	Random Forest	100.00	71.50
1	Gaussian Naive Bayes	53.59	57.51

RESULTS AND DISCUSSION :

The project's main goal is to accurately classify patients as having liver disease or not. The Conclusion from the Models (Logistic Regression, Gaussian Naive Bayes, Random Forest) **is that the**

Logistic Regression perform the best on this dataset

CONCLUSION :

Thus we conclude a decision tree is. So, after a long journey of data visulaisation, data cleaning, data modelling etc., we have finally got our model that we can use.

REFERENCES :

[1] Namrata S Gupta, Bijendra S Agrawal, Rajkumar M. Chauhan, ìSurvey On Clustering Technique of Data Miningî, American International Journal of Research in Science, Technology, Engineering & Mathematics,ISSN:2328-3491.

[2] Malwindersingh, Meenakshibansal ,î A Survey on Various K- Means algorithms for Clusteringî, IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015.

[3] Amandeep Kaur Mann, Navneet Kaur Mann, ìReview Paper On Clustering Techniques, Global Journal Of Computer Science And Technology Software & Data Engineering, VOL. 13 ,2013.

[4] Leonid Churilov. Adyl Bagirov, Daniel Schwarta, Kate Smith, Michael Dally , Journal of management information system : 2005, Data mining with combined use of optimization techniques and self organizing maps for improving risk grouping rules : application to prostate cancer patients

[5] Anthony Danna, Oscar H. Gandy, Journal of business ethics : 2002, All that glitters is not gold : Digging Beneath the surface of data mining

[6] AC Yeo, KA Smith, RJ Willis and M Brooks, Journal of the operation research society : 2002 , A mathematical programming approach to optimize insurance premium pricing within a data minning framework.

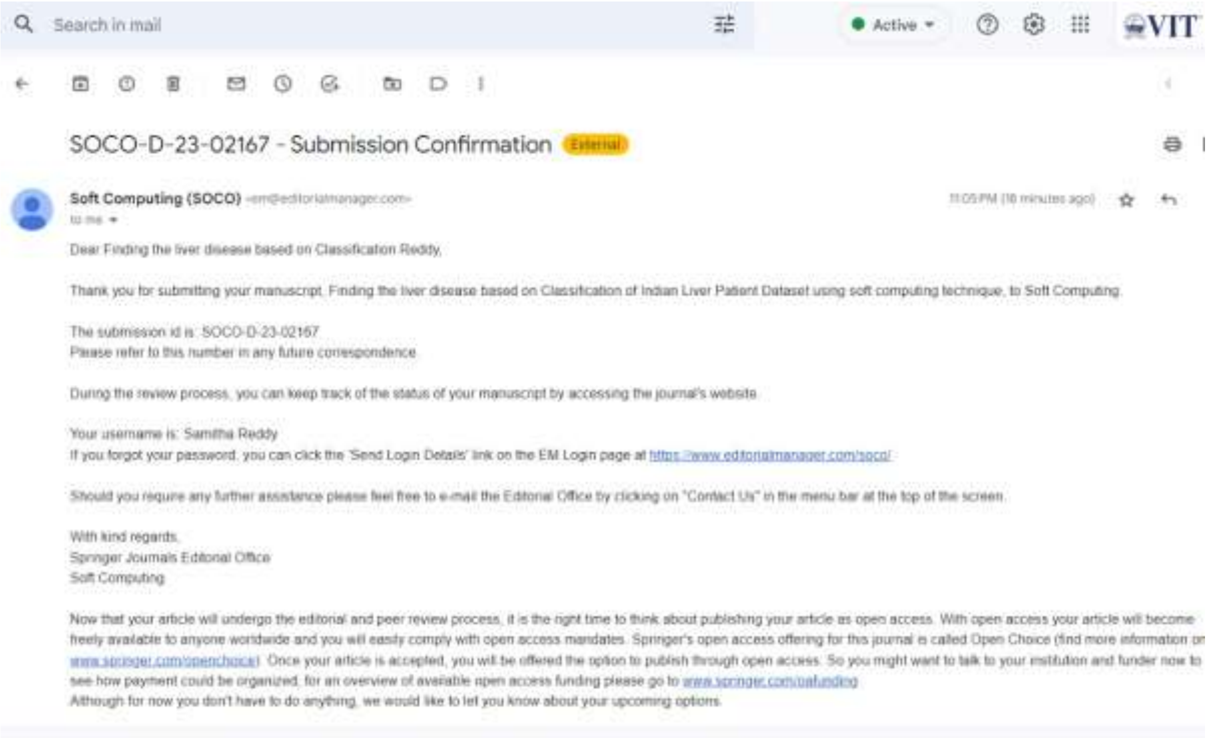
ABOUT AUTHORS :

Dr. S. AYYASAMY obtained his bachelor’s degree in Electronics and Communication Engineering from “Maharaja Engineering College, Avinashi” under Bharathiyar University and master’s degree in Computer Science and Engineering from “PSG College of Technology, Peelamedu, Coimbatore.” under Bharathiyar University. He completed his Ph. D in Computer Science and Engineering under Anna University, Chennai. He has 21 years of teaching experience in various reputed institutions. He is currently working as Professor in School of Computer Science and Engineering, VIT University, Vellore Campus, Tamil Nadu India. He has authored more than 30 research papers in reputed international journals and well-respected international conferences. He has conducted 05 International National Conferences as a convenor. He has been working as Reviewer for reputed international journals. His Research areas of interest are Peer to Peer Networks, Overlay Networks, Soft Computing, Machine learning and Artificial Intelligence.

Ms. Abhinaya is pursuing her Int. MTech CSE (Computer Science and Engineering), 2020 – 2025, in Vellore Institute of Technology, Tamil Nadu, India

Ms. Samitha is pursuing her Int. MTech CSE (Computer Science and Engineering), 2020 – 2025, in Vellore Institute of Technology, Tamil Nadu, India

Ms. Thejashwini is pursuing her Int. MTech CSE (Computer Science and Engineering), 2020 – 2025, in Vellore Institute of Technology, Tamil Nadu, India



SOCO-D-23-02167 - Submission Confirmation

Soft Computing (SOCO) <em@editorialmanager.com>

11:05 PM (18 minutes ago)

Dear Finding the liver disease based on Classification Reddy,

Thank you for submitting your manuscript, Finding the liver disease based on Classification of Indian Liver Patient Dataset using soft computing technique, to Soft Computing.

The submission id is: SOCO-D-23-02167
Please refer to this number in any future correspondence.

During the review process, you can keep track of the status of your manuscript by accessing the journal's website.

Your username is: Samitha Reddy
If you forgot your password, you can click the 'Send Login Details' link on the EM Login page at https://www.editorialmanager.com/soco/

Should you require any further assistance please feel free to e-mail the Editorial Office by clicking on "Contact Us" in the menu bar at the top of the screen.

With kind regards,
Springer Journals Editorial Office
Soft Computing

Now that your article will undergo the editorial and peer review process, it is the right time to think about publishing your article as open access. With open access your article will become freely available to anyone worldwide and you will easily comply with open access mandates. Springer's open access offering for this journal is called Open Choice (find more information on www.springer.com/openchoice). Once your article is accepted, you will be offered the option to publish through open access. So you might want to talk to your institution and funder now to see how payment could be organized; for an overview of available open access funding please go to www.springer.com/oafunding.
Although for now you don't have to do anything, we would like to let you know about your upcoming options.