# 1. INTRODUCTION

## 1.1 Overview

### A Review of Liver Patient Analysis Methods Using Machine Learning

Liver diseases averts the normal function of the liver. This disease is caused by an assortment of elements
that harm the liver. Diagnosis of liver infection at the preliminary stage is important for better treatment. In
today's scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples. This disease diagnosis is very
costly and complicated. Therefore, the goal of this work is to evaluate the performance of different Machine
Learning algorithms in order to reduce the high cost of liver disease diagnosis. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the
disease within a short duration of time. In this project we will analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for

determining the liver disease. This project compares various classification algorithms such as Random

Forest, Logistic Regression, KNN and ANN Algorithm with an aim to identify the best technique. Based on
this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further
utilised in the prediction of liver disease and can be recommended to the user.

## 1.2 Purpose

Liver function tests are blood tests used to help diagnose and monitor liver disease or damage. The tests measure the levels of certain enzymes and proteins in your blood.

Some of these tests measure how well the liver is performing its normal functions of producing protein and clearing bilirubin, a blood waste product. Other liver function tests measure enzymes that liver cells release in response to damage or disease.

Liver function tests can be used to:

Screen for liver infections, such as hepatitis

Monitor the progression of a disease, such as viral or alcoholic hepatitis, and determine how well a treatment is working

Measure the severity of a disease, particularly scarring of the liver (cirrhosis)

Monitor possible side effects of medications

Liver function tests check the levels of certain enzymes and proteins in your blood. Levels that are higher or lower than normal can indicate liver problems. Some common liver function tests include:

- Alanine transaminase (ALT). ALT is an enzyme found in the liver that helps convert proteins into energy for the liver cells. When the liver is damaged, ALT is released into the bloodstream and levels increase.
- Aspartate transaminase (AST). AST is an enzyme that helps metabolize amino acids. Like ALT, AST is normally present in blood at low levels. An increase in AST levels may indicate liver damage, disease or muscle damage.
- Alkaline phosphatase (ALP). ALP is an enzyme found in the liver and bone and is important for breaking down proteins. Higher-than-normal levels of ALP may indicate liver damage or disease, such as a blocked bile duct, or certain bone diseases.
- Albumin and total protein. Albumin is one of several proteins made in the liver. Your body needs these proteins to fight infections and to perform other functions. Lower-than-normal levels of albumin and total protein may indicate liver damage or disease.
- Bilirubin. Bilirubin is a substance produced during the normal breakdown of red blood cells. Bilirubin passes through the liver and is excreted in stool. Elevated levels of bilirubin (jaundice) might indicate liver damage or disease or certain types of anemia.
- Gamma-glutamyltransferase (GGT). GGT is an enzyme in the blood. Higher-than-normal levels may indicate liver or bile duct damage.
- L-lactate dehydrogenase (LD). LD is an enzyme found in the liver. Elevated levels may indicate liver damage but can be elevated in many other disorders.
- Prothrombin time (PT). PT is the time it takes your blood to clot. Increased PT may indicate liver damage but can also be elevated if you're taking certain blood-thinning drugs, such as warfarin.

**4. ADVANTAGES & DISADVANTAGES**

**Advantages :**

To discover hidden knowledge and patterns from the liver disease patient dataset: Investigating current data is important as the knowledge and patterns discovered act as a base for this project. The data used to train the model has to be clean with no noise in order to express the result with a higher accuracy. This objective is fulfilled using the Liver Disease Patient Dataset that is obtained from Kaggle.

To build a classification model for predicting the possibility of liver patients: A classification model is developed using the RStudio software to determine the target class, with the possibilities set to ‘Liver Patient' and ‘Non-Liver Patient’. Classification algorithms, which are used to construct classification models from a dataset are one of the most important approaches in data mining. To evaluate the model's performance, the Naïve Bayes algorithm and the Decision Tree algorithm will be implemented.

To develop an effective prediction system to identify liver patients: A working prototype is developed using programming languages such as PHP, JavaScript and HTML to aid general practitioners in identifying a liver patient and if they should be referred to a hepatologist.

To perform suitable testing to ensure the effectiveness of the proposed system: Suitable testing to ensure the effectiveness of the proposed system is important in order to ensure that the system fits its required specifications and functions. This final phase aid in identifying the defects in the system. The system can be regarded to be performing efficiently if the testing works true to form.

The rest of the sections of the paper are organized as follows. Section 2 presents the literature survey of liver diseases, Sect. 3 includes the details of the proposed methodology, Sect. 4 includes a description of the dataset, Results, and discussions in Sect. 5 and finally, conclusion is included in Sect. 6.

**Disadvantages :**

The challenges with the reliability of healthcare data and the constraints of predictive modelling are causing data mining efforts to fail. As Big Data has gained interest in recent years, involvement in the implementation of data mining techniques and methods to analyze healthcare data has emerged [28,29,30,31]. According to research [4], data mining in healthcare are facing limitations such as medical data reliability, data sharing across healthcare organizations, and improper modelling leading to untrue predictions.

The challenges with the reliability of healthcare data and the constraints of predictive modelling are causing data mining efforts to fail. As Big Data has gained interest in recent years, involvement in the implementation of data mining techniques and methods to analyze healthcare data has emerged [28,29,30,31]. According to research [4], data mining in healthcare are facing limitations such as medical data reliability, data sharing across healthcare organizations, and improper modelling leading to untrue predictions.

Data mining approaches have already been proved to be effective in eliciting previously untapped relevant insights from huge medical datasets. This section highlights previous research that employs classification analysis for a variety of objectives in the healthcare field. In the medical sector, classification tasks have been performed for a variety of motives.

According to research [11], current technological breakthroughs were used to construct prediction models for breast cancer survival. The prediction models were developed using three common data mining algorithms which are Naïve Bayes, RBF Network, and J48. These classification techniques were chosen by the authors because they are frequently used in research and have the ability to deliver valuable outcomes. Furthermore, these techniques can generate classification models in a variety of ways, increasing the likelihoods of obtaining a prediction model with high classification accuracy. In the pre-processing stage, the instances with missing values were omitted from the dataset in order to create a fresh dataset. The WEKA version 3.6.9 tool was used in this study to evaluate the performance of data mining techniques deployed to a medical dataset. The measurements of model performance are discussed which serve as the foundation for comparing the

efficiency and accuracy of different methodologies. The Naïve Bayes model was shown to be the most accurate predictor on the holdout sample, with 97.36% accuracy, followed by RBF Network with 96.77% accuracy and J48 with 93.41% accuracy.

**5. APPLICATIONS :**

In India, delays in diagnosing diseases are a major problem due to a lack of medical professionals. The typical scenario, which is mainly in rural and slightly urban areas:

1. A patient who sees a doctor with certain symptoms.
2. The doctor will perform some tests, such as blood and urine tests, depending on the symptoms.
3. The patient undergoes the above tests in the analytical laboratory.
4. The patient takes the reports back to the hospital, where they are examined and diagnosed.

The goal of this project is to reduce some of the delays caused by unnecessary detours between the hospital and the pathology laboratory. Historically, work has been done to detect the onset of heart disease, such as Parkinson's, and machine learning algorithms have been developed to predict liver disease in patients based on a variety of characteristics.

"Considering the data set containing the various attributes of 584 Indian patients, use the functions in the data set and determine a controlled classification algorithm to determine whether a person is suffering from liver disease. This data set contains 416 liver recordings and 167 non-liver recordings. collected in northeastern Andhra Pradesh. This data set contains records of 441 male patients and 142 female patients. Each patient over the age of 89 is "90" years old.

This seems to be a classic example of controlled learning. We are given a fixed number of functions for each data point, and our goal is to teach different controlled learning algorithms based on this data, so that when a new data point appears, our best-performing classifier can be used. information point as a positive or negative example. Detailed information on the number and types of algorithms used for training is contained in the "Algorithms and Techniques" section of the "Analysis" section.

**6. CONCLUSION :**

The proposed technique offers a variety of data visualization and user interface options, and this type of platform can be used as an early diagnosis tool for liver-related disorders in the healthcare sector. This study suggests a machine learning-based technique for predicting liver disease. The framework includes a user interface via which healthcare providers can enter patient information.

Initially, the data set was studied and prepared for inclusion in the classifiers. This was achieved by removing some rows containing zero values, modifying some columns indicating the skewness, and using appropriate conversion techniques (a hot coding) to make the labels more useful for classification purposes. The performance indicators for which the models will be evaluated have been resolved. The data set was then divided into a reading and testing package.

First, a simple predictive and base model ("Logistic Regression") was developed in the data set to determine the value of the base accuracy. The biggest challenge in implementing this project was in two areas: defining learning algorithms and selecting the appropriate parameters for precise

configuration. Initially, making a decision on 3 or 4 methods out of the many choices available at sklearn was very tedious.

Algorithms and Techniques used to develop this Liver Patient Analysis Machine Learning Project are

1. Random Forest Classifier:

2. Gaussian Naive Bayes Classifier

3. Logistic Regression:

**7. FUTURE SCOPE :**

This research work evaluates the performance of ensemble-based machine learning algorithms on the ILPD (Indian Liver Patient Dataset) and compares their results. The Ensemble technique is a unique approach in which we combine multiple machine learning models of the same or different types such as decision tree, logistic regression, support vector machines, etc., to carry out prediction [31]. The models used in ensemble models are called base estimators or base learners. There are many reasons to use ensemble models over traditional models. A few reasons are mentioned below.

- Performance: A single model may not be able to give reliable results. Combining multiple models helps to increase prediction accuracy [32].
- Robustness: An ensemble helps in reducing the spread in the average performance of the machine learning model [32].
- Low variance: Ensembles help in reducing the variance (error) of the prediction by combining multiple models [32].

One model might not be able to forecast a dataset's outcomes to the best of its ability. Therefore, simple machine learning models have limitations, and it is difficult to create a model with great accuracy [31]. If multiple models are combined, then the accuracy is boosted. Ensembles work on the mechanism of aggregation of output from individual models in such a way that model error is reduced, and generalization is maintained [31]. The algorithms employed in this research work have been thoroughly discussed in detail in the following sections

> *Gradient Boosting Classification Algorithm to Predict Liver Disease*
>
> *In order to create a powerful regression or classification model, the gradient boosting classifier combines a number of weak learning models [33]. Decision trees are frequently used in gradient boosting. Due to their proficiency in classifying challenging datasets, gradient boosting models are becoming more and more popular, and have recently prevailed in a number of Kaggle data science competitions [33].*

*Gradient boosting classification has three main components as shown in Algorithm 1.*

*Loss function: It determines how well a model is doing a prediction. More loss means the model could do better and vice versa [34]. Gradient descent is used to minimize this loss function value.*

*Weak learner: A weak learner classifies data very poorly and can be comparable to random guessing. It has a high rate of errors. Usually, decision trees are used in this [34].*

*Additive model: In this approach, trees are added iteratively and sequentially one at a time. After each iteration, the model is usually closer to the actual target [34].*

*Algorithm 1 Gradient Boosting to Predict Liver Disease*

*XGBoosting Classification Algorithm to Predict Liver Disease Similar to the gradient boosting algorithm, XGBoost uses gradient descent to enhance weak learners. However, XGBoost improves due to system optimization and algorithmic upgrades [35]. The system optimization applied in XGBoost are Parallelization, Tree Pruning and Hardware. The algorithmic enhancements applied in XGBoost are weighted quantile sketch, Regularization, Cross-validation, and Sparsity awareness.*

### *Bagging Classification Algorithm to Predict Liver Disease*

Bagging (bootstrap aggregation) is a classification technique that reduces the variance of prediction by taking the average of multiple predictions together [36]. Subsets called bootstrap samples (samples with replacement) are created from the main dataset, and the different base estimators are trained on these subsets [36]. This is called row sampling with replacement. The voting method (majority) is used in the case of classification for aggregating the prediction of different classifiers [36]. The variance decreases and the model's performance rises by averaging the results [36]. Base classifiers such as decision tree SVM, etc. can be used. Algorithm 2 depicts the bagging approach to predict liver disease.

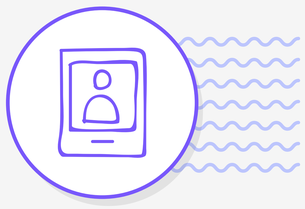
*Random Forest Classification Algorithm to Predict Liver Disease Random forest classification algorithm is a type of Bagging method in which all the base learners are decision trees, and data samples are split by replacement. Random feature sampling is also applied. The best split approach is used while splitting the data. Decision tree is a weak learner and using multiple decision trees together has helped gain better results. Algorithm 3 depicts the random forest classification approach to predict liver disease.*

*Extra Tree Classification Algorithm to Predict Liver Disease Extra tree classification algorithm is an extended version of random forest with some variations. Similar to random forest, all the base learners are decision trees, but data samples are split randomly without replacement. Hence, instead of using the best split random split approach is used. Features are split randomly similar to random forest. This algorithm has given better results on noisy datasets compared with the random forest approach. Algorithm 4 depicts the extra tree classification algorithm to predict liver disease.*

*Ensemble Stacking Classification Algorithm to Predict Liver Disease Stacking algorithms base estimators use the entire training dataset during training [36]. Once these base learners are trained, a meta-learner is assembled from the different models, and the base learner's output is used for the training of the meta-learners [36]. A heterogeneous ensemble is created by this approach as the base learners are usually different algorithms [36]. This work uses ensembles models as the base model for stacking as shown in Algorithm 5.*

*In the above subsections, various feature scaling, feature selection methods, and machine learning algorithms are discussed. First, the data are split into training and testing so that every model receives the same train test split. Then, for each of the six algorithms, a default model is trained, and the best pair of feature scaling and feature selection combination is found. This is carried out by training and checking the model's training accuracy on all combinations of feature scaling and selection pairs. In order to obtain the best feature scaling and selection pair, the optimal hyper-parameters are obtained using grid search with 10-fold cross validation. GridSearchCV from the sklearn library was used for this purpose. It uses all the specified hyper-parameters in various combinations and then calculates the performance for each. The best value for the hyper-parameters is then chosen. We performed training on hyper-parameters obtained from grid search, as well as default hyper-parameters. The best out of the two was chosen for comparison. The hyperparameter optimization carried out for all the models is given in Table 2. Finally, the models are trained by passing this list of optimal parameters to each model. Parameters such as the number of estimators, learning rate, etc., are passed to the models. The best result for each model evaluated on the test set (external validation) is then stored. This proposed method is named as enhanced preprocessing.*

Template

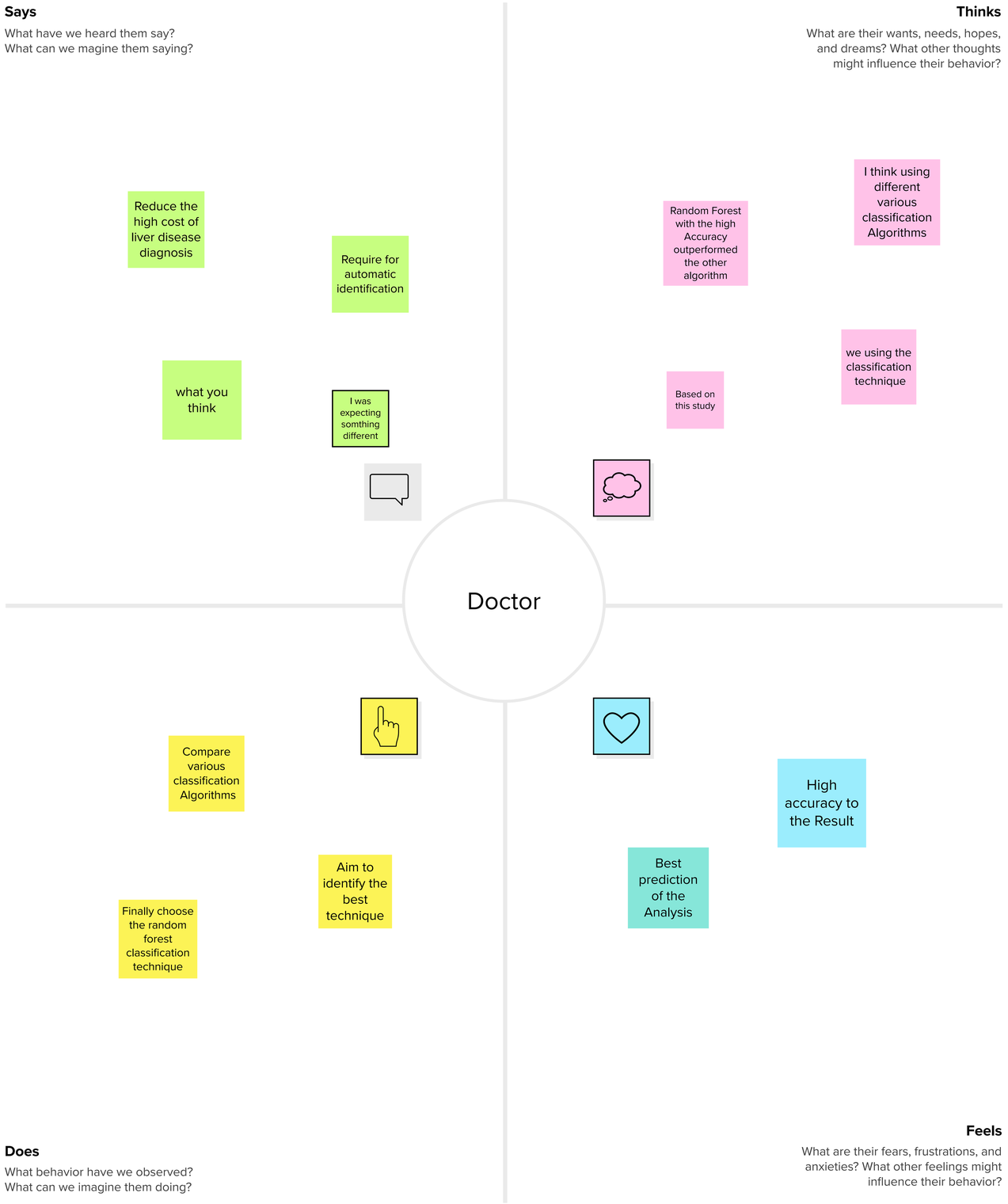


# Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

## Build empathy

The information you add here should be representative of the observations and research you've done about your users.




Share template feedback

## Need some inspiration?

See a finished version of this template to kickstart your work.

Open example

**2.1 Ideation & Brainstorming map :**

- Says:

1. Required for automation identification.

2. Reduce the high cost of liver disease diagnosis.

3. I was excepted something different , what you think.

- Think :

1. We using classification technique.

2. I think using different various classification Algorithms.

3. Based on this study , Random forest with the high accuracy outperformed the other algorithms.

- Does:

1. This Project compare various classification Algorithms such as Random forest , Logistic regression KNN and ANN Algorithms.

2. Aim to identify the best Technique.

3. Finally i choose the Random forest classification algorithm technique to build the project.

- Feel:

1. Best prediction of the Analysis.

2. High accuracy to the results.

Reduce the high cost of liver disease diagnosis

Require for automatic identification

what you think

I was expecting somthing different

Random Forest with the high Accuracy outperformed the other algorithm

I think using different various classification Algorithms

we using the classification technique

Based on this study

Doctor

Compare various classification Algorithms

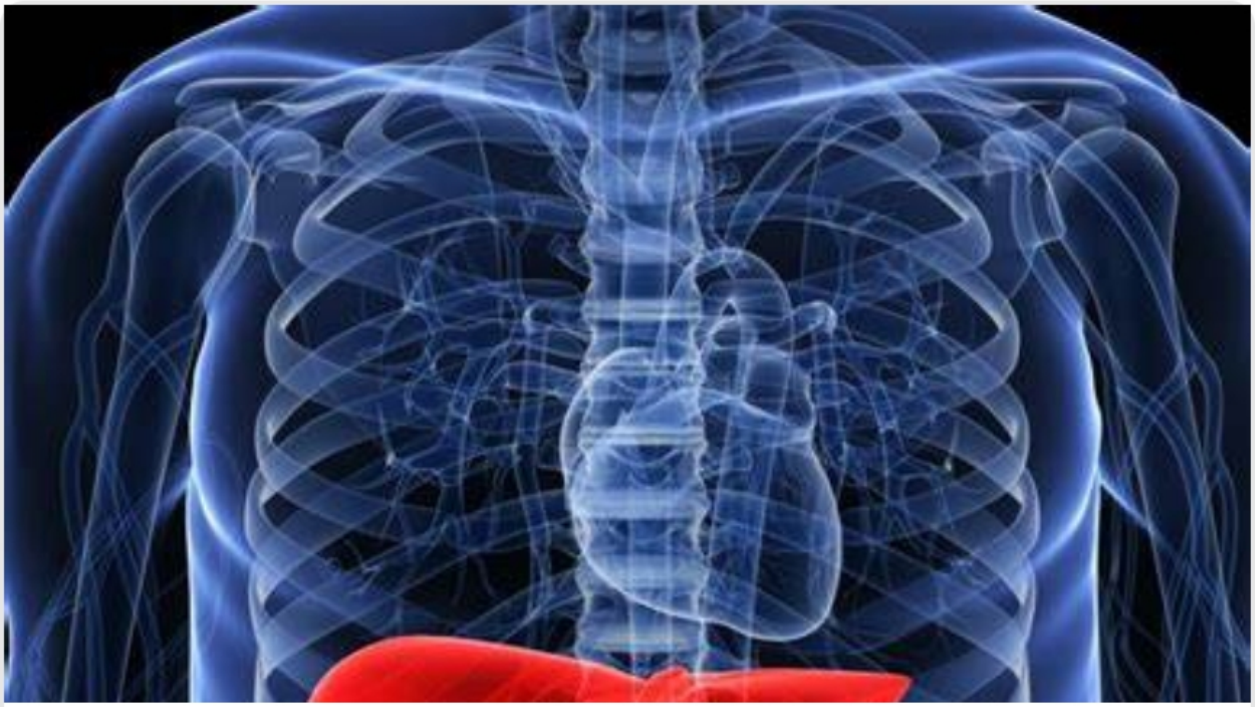
Aim to identify the best technique

Finally choose the random forest classification technique

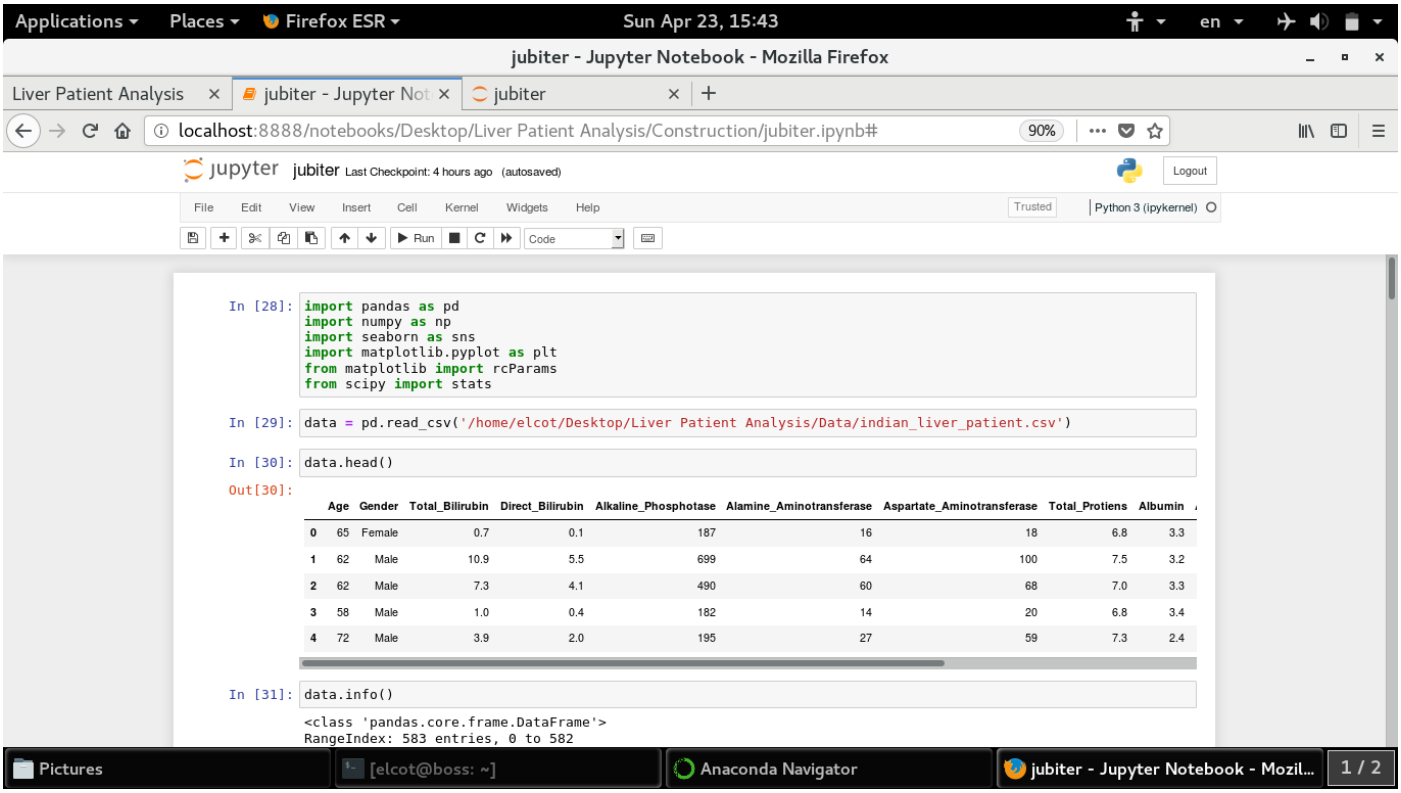
High accuracy to the Result

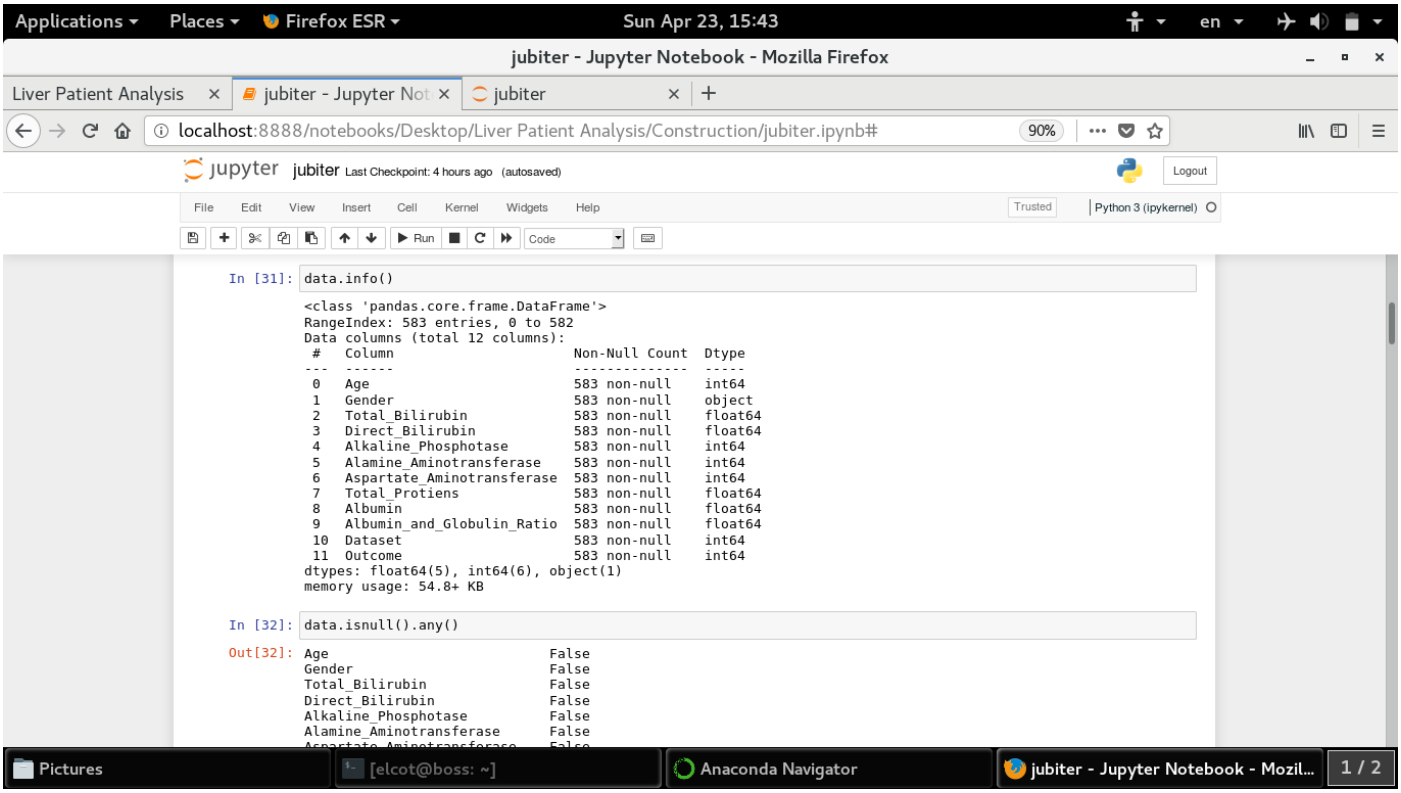
Best prediction of the Analysis

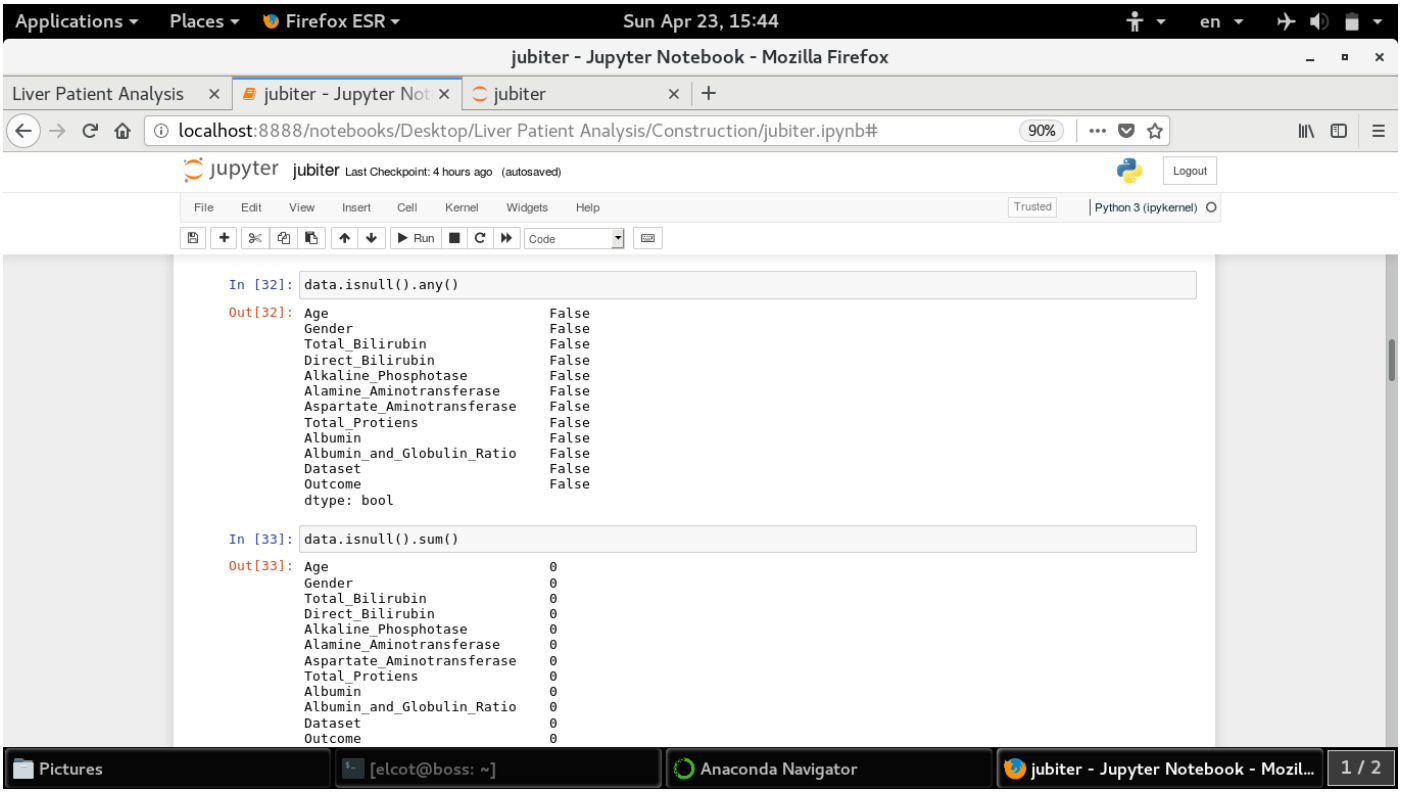
## 8. APPENDIX :

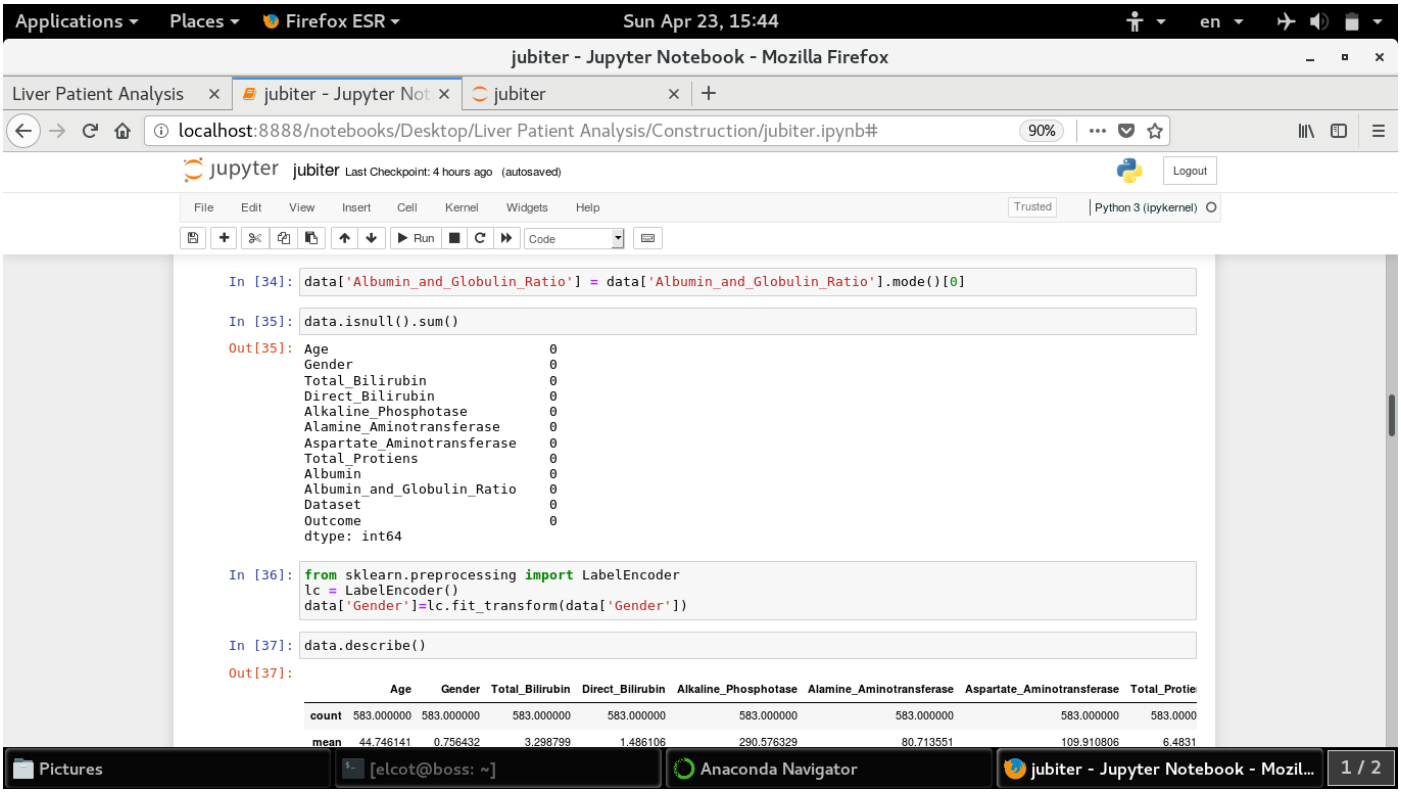


# A Review Of Liver Patient Analysis Methods Using Machine Learning


Applications
Places
Firefox ESR
Sun Apr 23, 15:43
en
jubiter - Jupyter Notebook - Mozilla Firefox
Liver Patient Analysis
jubiter - Jupyter Not
jubiter
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb#
90%
jupyter
jubiter Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [28]:
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams
from scipy import stats
In [29]:
data = pd.read_csv('/home/elcot/Desktop/Liver Patient Analysis/Data/indian_liver_patient.csv')
In [30]:
data.head()
Out[30]:
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin
0 65 Female 0.7 0.1 187 16 18 6.8 3.3
1 62 Male 10.9 5.5 699 64 100 7.5 3.2
2 62 Male 7.3 4.1 490 60 68 7.0 3.3
3 58 Male 1.0 0.4 182 14 20 6.8 3.4
4 72 Male 3.9 2.0 195 27 59 7.3 2.4
In [31]:
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Pictures
[elcot@boss: ~]
Anaconda Navigator
jubiter - Jupyter Notebook - Mozil...
1 / 2


Applications
Places
Firefox ESR
Sun Apr 23, 15:43
en
jubiter - Jupyter Notebook - Mozilla Firefox
Liver Patient Analysis
jubiter - Jupyter Not
jubiter
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb#
90%
jupyter
jubiter Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [31]:
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 12 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 583 non-null float64
10 Dataset 583 non-null int64
11 Outcome 583 non-null int64
dtypes: float64(5), int64(6), object(1)
memory usage: 54.8+ KB
In [32]:
data.isnull().any()
Out[32]:
Age False
Gender False
Total_Bilirubin False
Direct_Bilirubin False
Alkaline_Phosphotase False
Alamine_Aminotransferase False
Pictures
[elcot@boss: ~]
Anaconda Navigator
jubiter - Jupyter Notebook - Mozil...
1 / 2


Applications
Places
Firefox ESR
Sun Apr 23, 15:44
en
jubiter - Jupyter Notebook - Mozilla Firefox
Liver Patient Analysis
jubiter - Jupyter Not
jubiter
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb#
90%
jupyter
jubiter Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [32]: data.isnull().any()
Out[32]:
Age False
Gender False
Total_Bilirubin False
Direct_Bilirubin False
Alkaline_Phosphotase False
Alamine_Aminotransferase False
Aspartate_Aminotransferase False
Total_Protiens False
Albumin False
Albumin_and_Globulin_Ratio False
Dataset False
Outcome False
dtype: bool
In [33]: data.isnull().sum()
Out[33]:
Age 0
Gender 0
Total_Bilirubin 0
Direct_Bilirubin 0
Alkaline_Phosphotase 0
Alamine_Aminotransferase 0
Aspartate_Aminotransferase 0
Total_Protiens 0
Albumin 0
Albumin_and_Globulin_Ratio 0
Dataset 0
Outcome 0
Pictures
[elcot@boss: ~]
Anaconda Navigator
jubiter - Jupyter Notebook - Mozil...
1 / 2


Applications
Places
Firefox ESR
Sun Apr 23, 15:44
en
jubiter - Jupyter Notebook - Mozilla Firefox
Liver Patient Analysis
jubiter - Jupyter Not
jubiter
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb#
90%
jupyter
jubiter Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [34]: data['Albumin_and_Globulin_Ratio'] = data['Albumin_and_Globulin_Ratio'].mode()[0]
In [35]: data.isnull().sum()
Out[35]:
Age 0
Gender 0
Total_Bilirubin 0
Direct_Bilirubin 0
Alkaline_Phosphotase 0
Alamine_Aminotransferase 0
Aspartate_Aminotransferase 0
Total_Protiens 0
Albumin 0
Albumin_and_Globulin_Ratio 0
Dataset 0
Outcome 0
dtype: int64
In [36]:
from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
data['Gender']=lc.fit_transform(data['Gender'])
In [37]: data.describe()
Out[37]:
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protie
count 583.000000 583.000000 583.000000 583.000000 583.000000 583.000000 583.000000 583.0000
mean 44.746141 0.756432 3.298799 1.486106 290.576329 80.713551 109.910806 6.4831
Pictures
[elcot@boss: ~]
Anaconda Navigator
jubiter - Jupyter Notebook - Mozil...
1 / 2

Applications Places Firefox ESR Sun Apr 23, 15:44 en

jubiter - Jupyter Notebook - Mozilla Firefox

Liver Patient Analysis | jubiter - Jupyter Not | jubiter

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb# 90%

jupyter jubiter Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [37]:
```
data.describe()
```

Out[37]:

| | Age | Gender | Total_Bilirubin | Direct_Bilirubin | Alkaline_Phosphotase | Alamine_Aminotransferase | Aspartate_Aminotransferase | Total_Protie |
|---|---|---|---|---|---|---|---|---|
| count | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.0000 |
| mean | 44.746141 | 0.756432 | 3.298799 | 1.486106 | 290.576329 | 80.713551 | 109.910806 | 6.4831 |
| std | 16.189833 | 0.429603 | 6.209522 | 2.808498 | 242.937989 | 182.620356 | 288.918529 | 1.0854 |
| min | 4.000000 | 0.000000 | 0.400000 | 0.100000 | 63.000000 | 10.000000 | 10.000000 | 2.7000 |
| 25% | 33.000000 | 1.000000 | 0.800000 | 0.200000 | 175.500000 | 23.000000 | 25.000000 | 5.8000 |
| 50% | 45.000000 | 1.000000 | 1.000000 | 0.300000 | 208.000000 | 35.000000 | 42.000000 | 6.6000 |
| 75% | 58.000000 | 1.000000 | 2.600000 | 1.300000 | 298.000000 | 60.500000 | 87.000000 | 7.2000 |
| max | 90.000000 | 1.000000 | 75.000000 | 19.700000 | 2110.000000 | 2000.000000 | 4929.000000 | 9.6000 |

In [38]:
```
sns.distplot(data['Age'])
plt.title('Age Distribution Graph')
plt.show()
```

```
/home/elcot/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future 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)
```

Pictures [elcot@boss: ~] Anaconda Navigator jubiter - Jupyter Notebook - Mozil... 1 / 2

Applications Places Firefox ESR Sun Apr 23, 15:44 en

jubiter - Jupyter Notebook - Mozilla Firefox

Liver Patient Analysis | jubiter - Jupyter Not | jubiter

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb# 90%

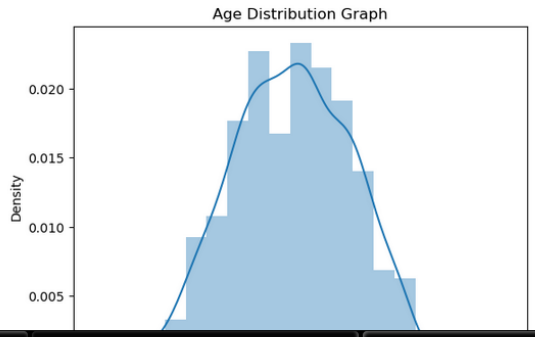
jupyter jubiter Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [38]:
```
sns.distplot(data['Age'])
plt.title('Age Distribution Graph')
plt.show()
```

```
/home/elcot/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future 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)
```




Pictures [elcot@boss: ~] Anaconda Navigator jubiter - Jupyter Notebook - Mozil... 1 / 2

Applications Places Firefox ESR Sun Apr 23, 15:45 en

jubiter - Jupyter Notebook - Mozilla Firefox

Liver Patient Analysis | jubiter - Jupyter Not | jubiter

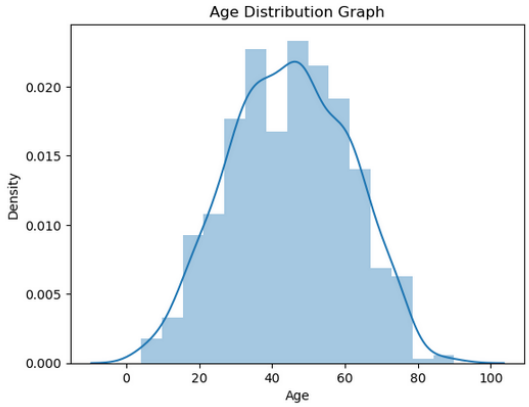
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb# 90%

jupyter jubiter Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
s a deprecated function and will be removed in a future version. Please adapt your code to use either `displ
ot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms)
.
  warnings.warn(msg, FutureWarning)
```




Pictures [elcot@boss: ~] Anaconda Navigator jubiter - Jupyter Notebook - Mozil... 1 / 2

Applications Places Firefox ESR Sun Apr 23, 15:45 en

jubiter - Jupyter Notebook - Mozilla Firefox

Liver Patient Analysis | jubiter - Jupyter Not | jubiter

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb# 80%

jupyter jubiter Last Checkpoint: 4 hours ago (autosaved) Logout

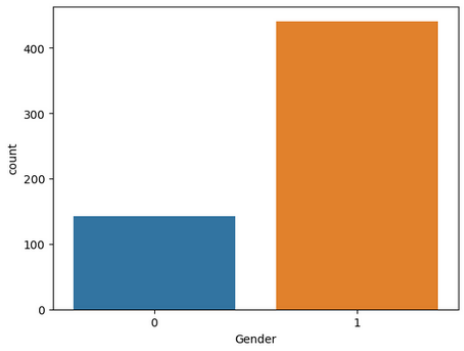
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [39]:
```
#sns.countplot(data['outcome'], hue=data['gender'])
sns.countplot(data['Gender'])
```

```
/home/elcot/anaconda3/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following
variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[39]: <AxesSubplot:xlabel='Gender', ylabel='count'>




Pictures [elcot@boss: ~] Anaconda Navigator jubiter - Jupyter Notebook - Mozil... 1 / 2

Applications Places Firefox ESR Sun Apr 23, 15:45 en

jubiter - Jupyter Notebook - Mozilla Firefox

Liver Patient Analysis | jubiter - Jupyter Not | jubiter

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb# 90%

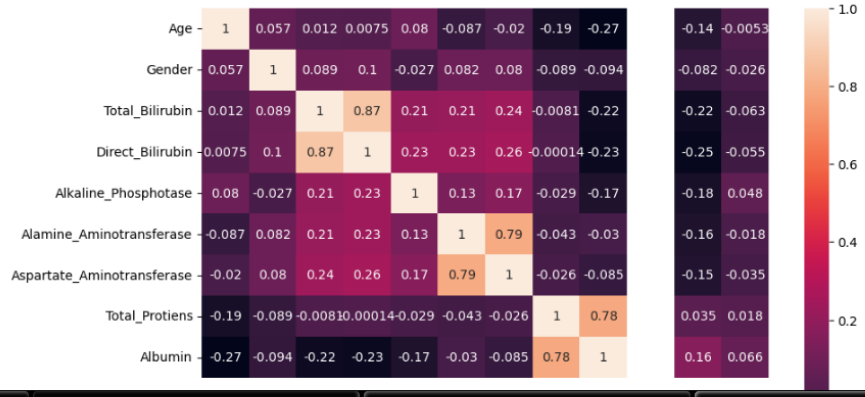
jupyter jubiter Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

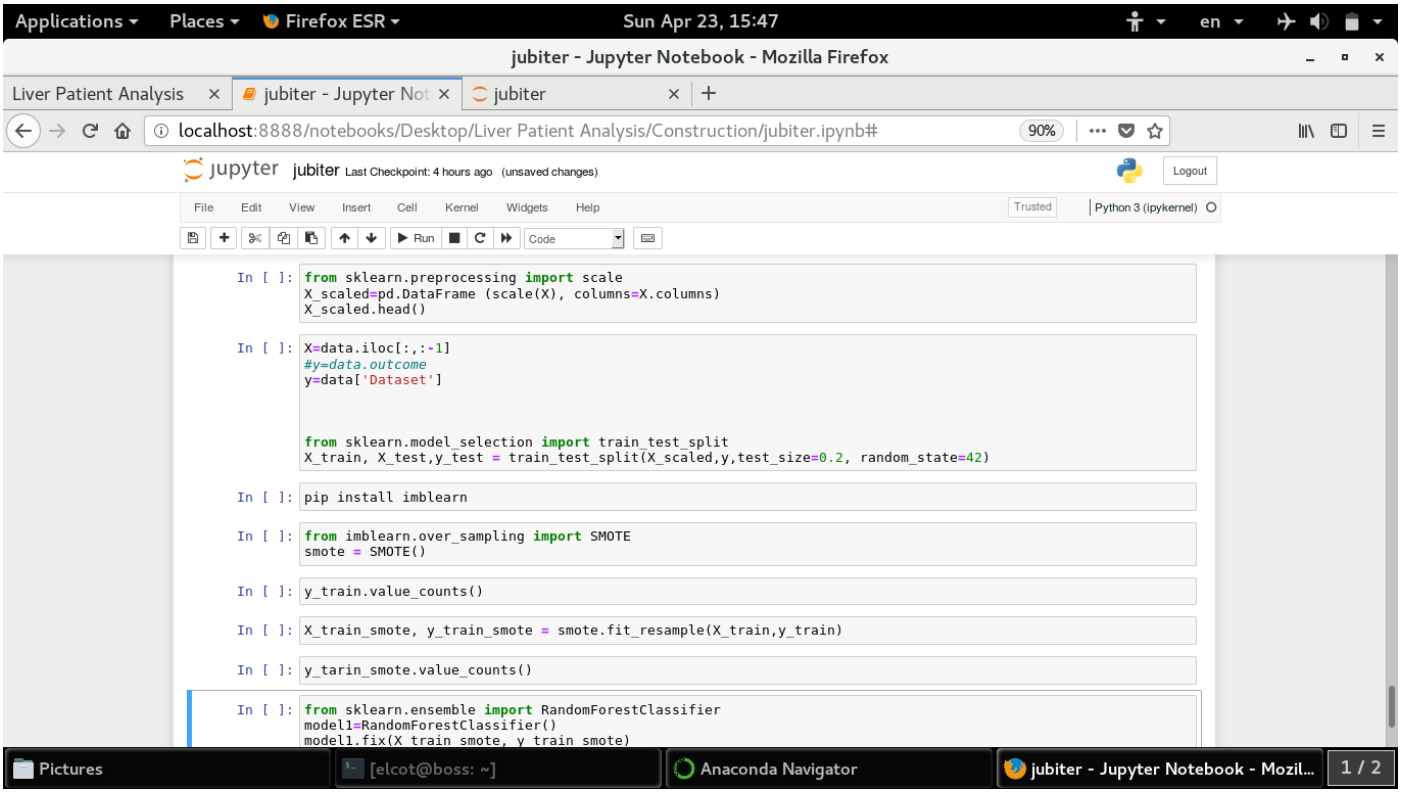
Run Code

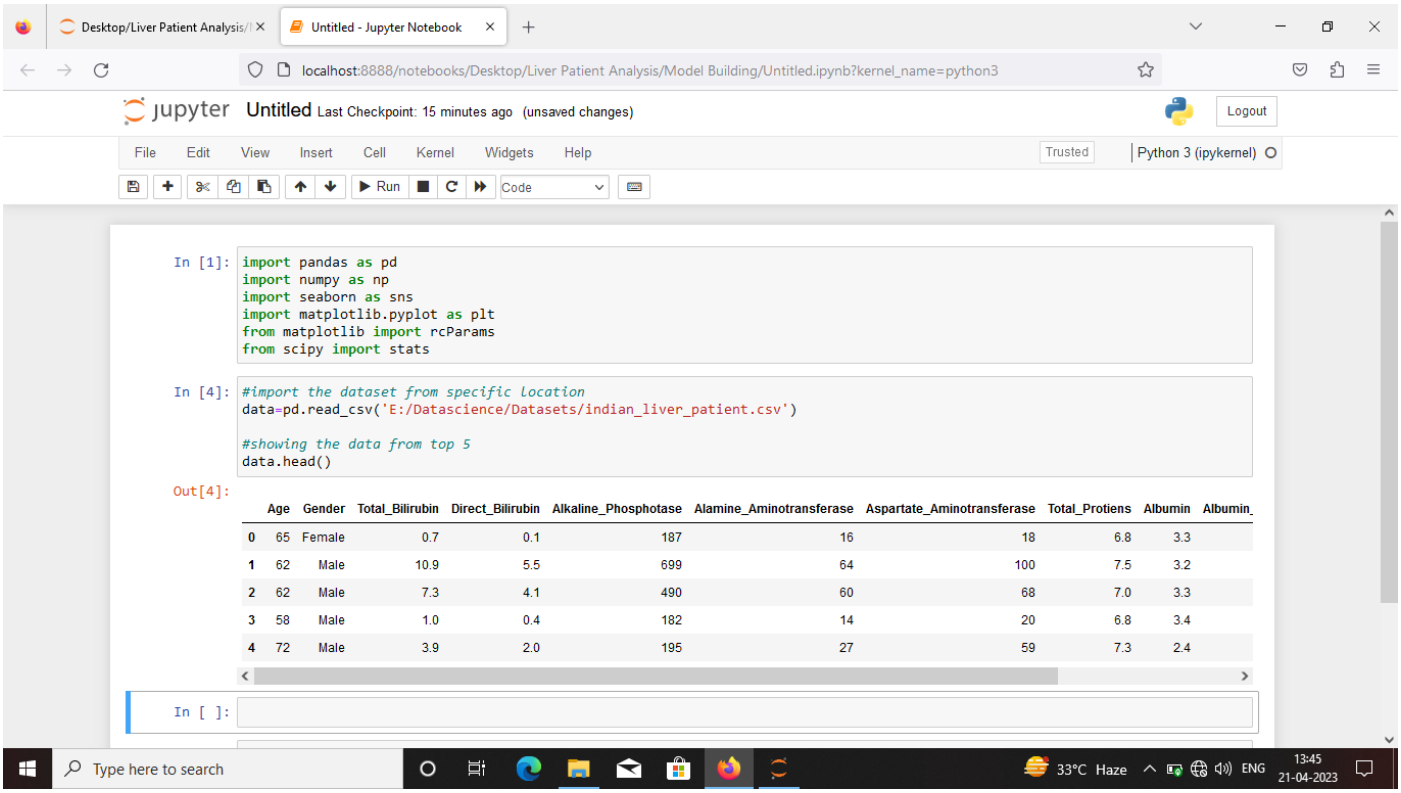
In [40]:
```
plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True)
```

Out[40]: <AxesSubplot:>




Pictures [elcot@boss: ~] Anaconda Navigator jubiter - Jupyter Notebook - Mozil... 1 / 2


Applications
Places
Firefox ESR
Sun Apr 23, 15:47
en
jubiter - Jupyter Notebook - Mozilla Firefox
Liver Patient Analysis
jubiter - Jupyter Not
jubiter
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Construction/jubiter.ipynb#
90%
jupyter
jubiter Last Checkpoint: 4 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [ ]:
from sklearn.preprocessing import scale
X_scaled=pd.DataFrame (scale(X), columns=X.columns)
X_scaled.head()
In [ ]:
X=data.iloc[:,:-1]
#y=data.outcome
y=data['Dataset']
from sklearn.model_selection import train_test_split
X_train, X_test,y_test = train_test_split(X_scaled,y,test_size=0.2, random_state=42)
In [ ]:
pip install imblearn
In [ ]:
from imblearn.over_sampling import SMOTE
smote = SMOTE()
In [ ]:
y_train.value_counts()
In [ ]:
X_train_smote, y_train_smote = smote.fit_resample(X_train,y_train)
In [ ]:
y_tarin_smote.value_counts()
In [ ]:
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fix(X_train_smote, y_train_smote)
Pictures
[elcot@boss: ~]
Anaconda Navigator
jubiter - Jupyter Notebook - Mozil...
1 / 2


Desktop/Liver Patient Analysis/I
Untitled - Jupyter Notebook
localhost:8888/notebooks/Desktop/Liver Patient Analysis/Model Building/Untitled.ipynb?kernel_name=python3
jupyter Untitled Last Checkpoint: 15 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams
from scipy import stats
In [4]:
#import the dataset from specific Location
data=pd.read_csv('E:/Datascience/Datasets/indian_liver_patient.csv')
#showing the data from top 5
data.head()
Out[4]:
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin Albumin_
0 65 Female 0.7 0.1 187 16 18 6.8 3.3
1 62 Male 10.9 5.5 699 64 100 7.5 3.2
2 62 Male 7.3 4.1 490 60 68 7.0 3.3
3 58 Male 1.0 0.4 182 14 20 6.8 3.4
4 72 Male 3.9 2.0 195 27 59 7.3 2.4
In [ ]:
Type here to search
33°C Haze
ENG
13:45
21-04-2023

Desktop/Liver Patient Analysis/I | Untitled - Jupyter Notebook

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Model Building/Untitled.ipynb?kernel_name=python3

jupyter **Untitled** Last Checkpoint: 11 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [18]:
```
sns.distplot(data['Age'])
plt.title('Age Distribution Graph')
plt.show()
```

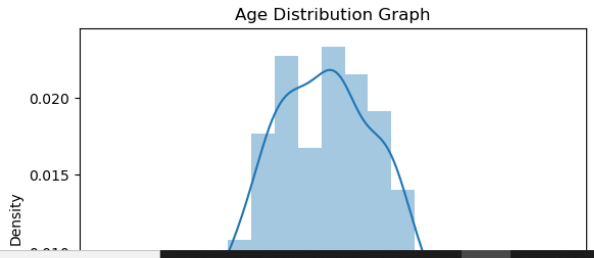
```
C:\Users\Black\AppData\Local\Temp\ipykernel_7900\2180260368.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['Age'])
```




Type here to search 33°C Haze ENG 03:24 22-04-2023

Desktop/Liver Patient Analysis/I | Untitled - Jupyter Notebook

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Model Building/Untitled.ipynb?kernel_name=python3

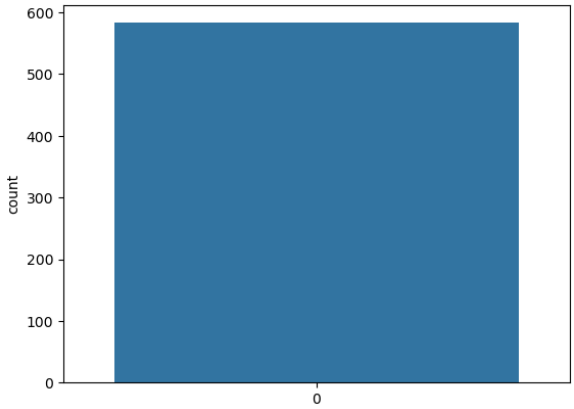
jupyter **Untitled** Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [74]:
```
#sns.countplot(data['gender'])
sns.countplot(data['Gender'])
```

Out[74]:
```
<Axes: ylabel='count'>
```




Type here to search 33°C Haze ENG 04:26 22-04-2023

Desktop/Liver Patient Analysis/I | Untitled - Jupyter Notebook

localhost:8888/notebooks/Desktop/Liver Patient Analysis/Model Building/Untitled.ipynb?kernel_name=python3

jupyter **Untitled** Last Checkpoint: an hour ago (unsaved changes) — Logout

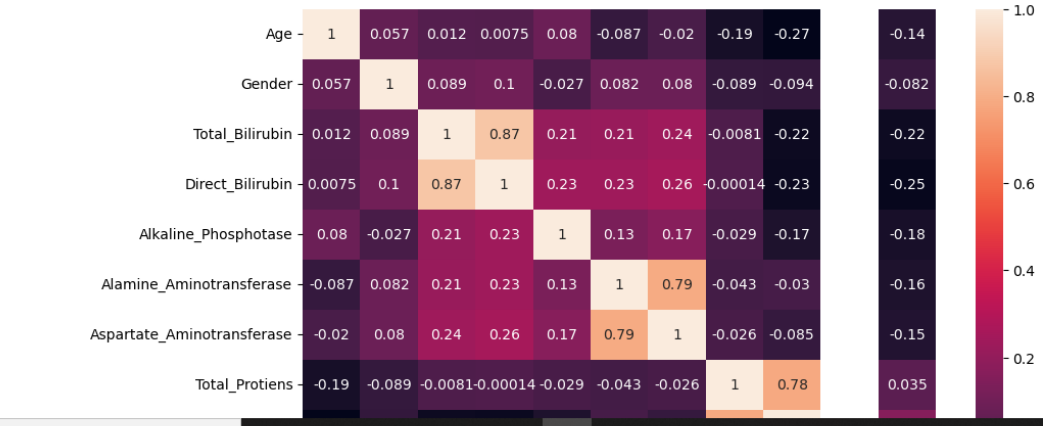
File Edit View Insert Cell Kernel Widgets Help — Trusted — Python 3 (ipykernel)

Run — Code

In [75]:
```
plt.figure(figsize=(10,7))

sns.heatmap(data.corr(),annot=True)
```

Out[75]: <Axes: >




Type here to search — 33°C Haze — ENG — 04:26 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

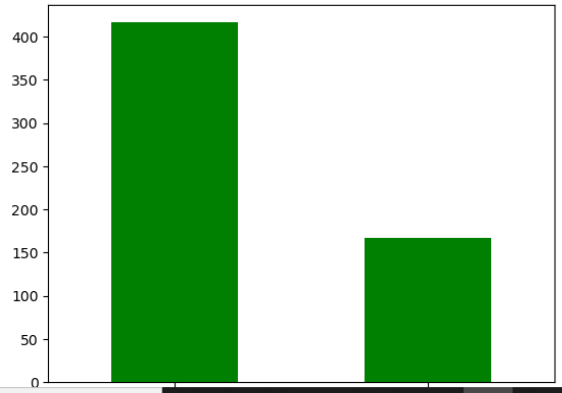
Run Code

```
memory usage: 50.2 KB
```

## # Step 4 : Data Visualization

In [19]:
```
liverdata['is_patient'].value_counts().plot.bar(color='green')
```

Out[19]: <Axes: >




Type here to search 36°C Mostly sunny ENG 13:25 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [23]:

```
#https://matplotlib.org/gallery/pie_and_polar_charts/pie_and_donut_labels.html#sphx-glr-gallery-pie-and-polar-charts
fig, ax=plt.subplots(figsize=(6,3),subplot_kw=dict(aspect='equal'))
yvalue_count=liverdata['is_patient'].value_counts()

data=[yvalue_count[1],yvalue_count[2]]
ingredients=["Liver patient",
             "Healthy"]

def func(pct,allvals):
    absolute=int(pct/100.*np.sum(allvals))
    return"{:.1f}%".format(pct,absolute)

wedges,texts,autotexts=ax.pie(data,autopct=lambda pct:func(pct,data),
                              textprops=dict(color="w"))

ax.legend(wedges,ingredients,
          title=None,
          loc="center left",
          bbox_to_anchor=(1,0))

plt.setp(autotexts,size=8,weight="bold")

ax.set_title("Healthy Vs Liver patient")

plt.show()
```




Type here to search

36°C Mostly sunny ENG 13:51 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

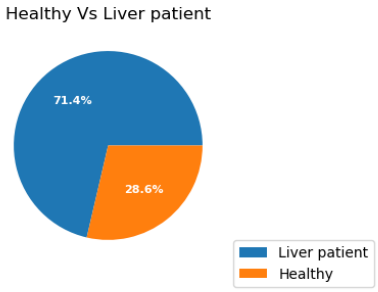
Run Code

```
ax.legend(wedges,ingredients,
          title=None,
          loc="center left",
          bbox_to_anchor=(1,0))

plt.setp(autotexts,size=8,weight="bold")

ax.set_title("Healthy Vs Liver patient")

plt.show()
```




In [ ]:

Type here to search 36°C Mostly sunny ENG 13:51 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: 2 hours ago (unsaved changes)

Logout

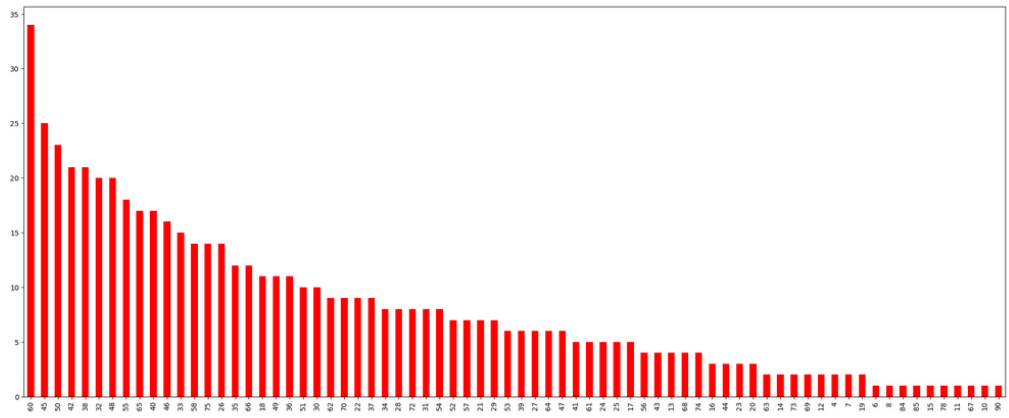
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [34]:
```
plt.figure(figsize=(25,10))
liverdata['age'].value_counts().plot.bar(color='red')
```

Out[34]: <Axes: >




Type here to search

36°C Mostly sunny

ENG

14:09
22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: 2 hours ago (unsaved changes)

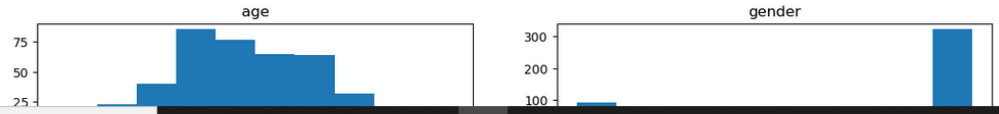
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

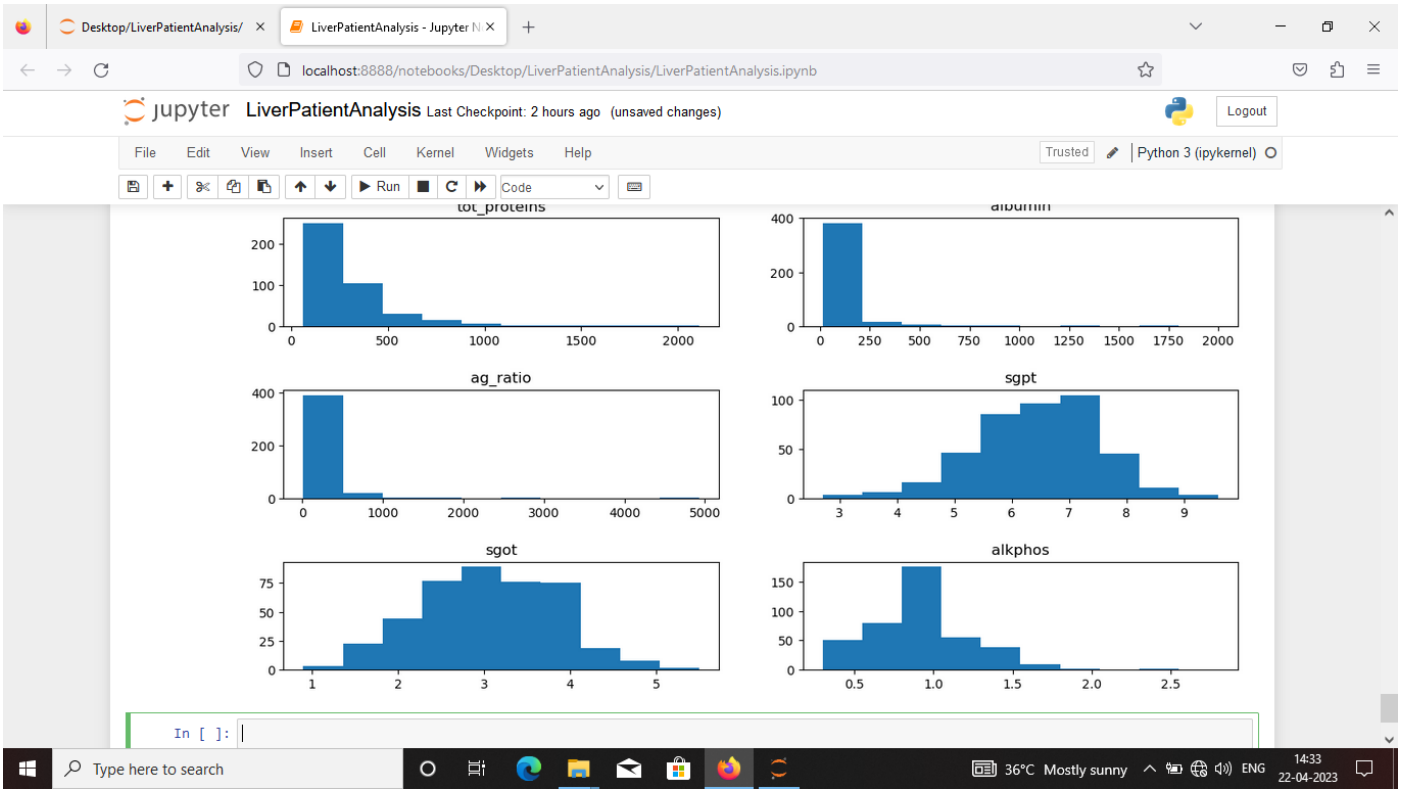
Run | Code

In [35]:
```
#Histogram of each feature
#Histogram of all columns when the Outcome is 1(liverpatient)
fig,ax=plt.subplots(nrows=5,ncols=2,figsize=(12,10))
fig.tight_layout(pad=3.0)
ax[0,0].set_title('age')
ax[0,0].hist(df.age[liverdata.is_patient==1]);
ax[0,1].set_title('gender')
ax[0,1].hist(df.gender[liverdata.is_patient==1]);
ax[1,0].set_title('tot_bilirubin')
ax[1,0].hist(df.tot_bilirubin[liverdata.is_patient==1]);
ax[1,1].set_title('direct_bilirubin')
ax[1,1].hist(df.direct_bilirubin[liverdata.is_patient==1]);
ax[2,0].set_title('tot_proteins')
ax[2,0].hist(df.tot_proteins[liverdata.is_patient==1]);
ax[2,1].set_title('albumin')
ax[2,1].hist(df.albumin[liverdata.is_patient==1]);
ax[3,0].set_title('ag_ratio')
ax[3,0].hist(df.ag_ratio[liverdata.is_patient==1]);
ax[3,1].set_title('sgpt')
ax[3,1].hist(df.sgpt[liverdata.is_patient==1]);
ax[4,0].set_title('sgot')
ax[4,0].hist(df.sgot[liverdata.is_patient==1]);
ax[4,1].set_title('alkphos')
ax[4,1].hist(df.alkphos[liverdata.is_patient==1]);
```




Type here to search

36°C Mostly sunny ENG 14:33 22-04-2023


Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: 2 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
tot_proteins
albumin
ag_ratio
sgpt
sgot
alkphos
In [ ]:
Type here to search
36°C Mostly sunny
ENG
14:33
22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

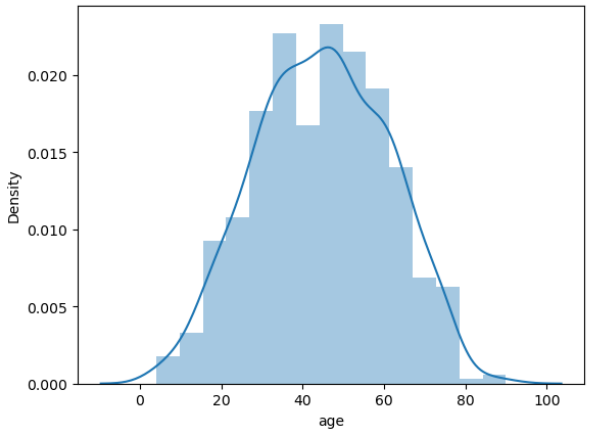
Trusted | Python 3 (ipykernel)

Run | Code

memory usage: 50.2 KB

In [30]:
```
sns.distplot(liverdata['age'])
```

Out[30]: <Axes: xlabel='age', ylabel='Density'>




Type here to search

36°C Mostly sunny ENG 15:41 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

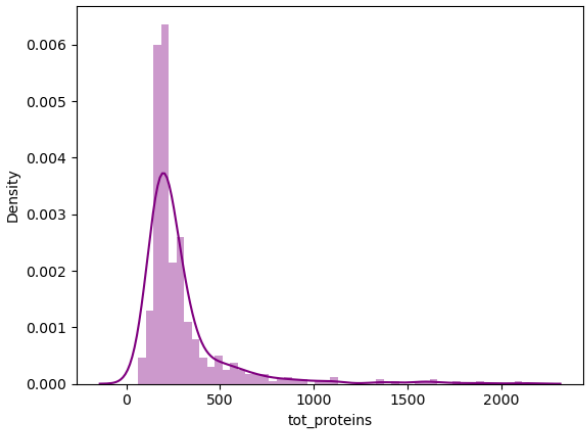
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [31]:
```
sns.distplot(liverdata['tot_proteins'],color='purple')
```

Out[31]:
```
<Axes: xlabel='tot_proteins', ylabel='Density'>
```




Type here to search

36°C Mostly sunny ENG 15:41 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

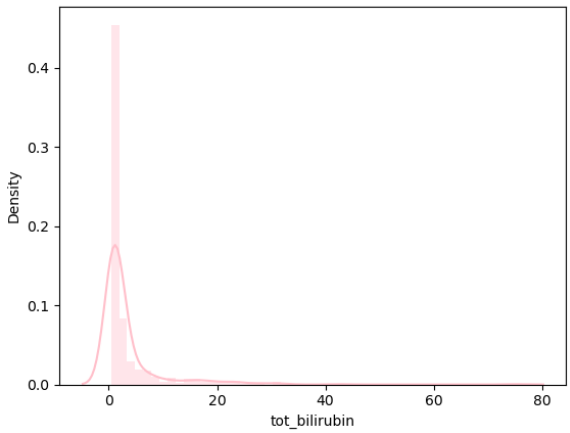
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [32]:
```
sns.distplot(liverdata['tot_bilirubin'],color='pink')#direct_bilirubin
```

Out[32]: <Axes: xlabel='tot_bilirubin', ylabel='Density'>




Type here to search

36°C Mostly sunny ENG 15:43 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

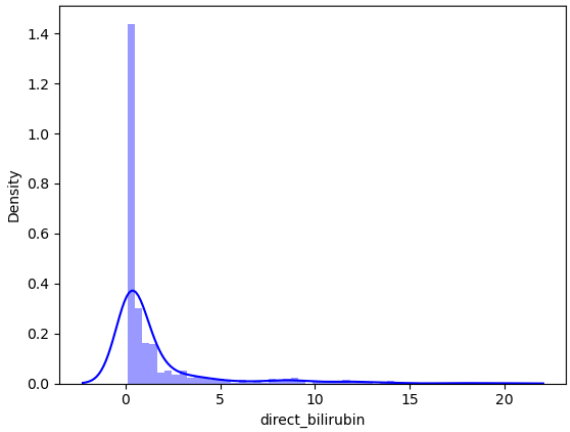
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [33]: sns.distplot(liverdata['direct_bilirubin'],color='blue')

Out[33]: <Axes: xlabel='direct_bilirubin', ylabel='Density'>




In [34]:
```
sns.distplot(liverdata['albumin'],color='red')
```

Type here to search

36°C Mostly sunny ENG 16:01 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

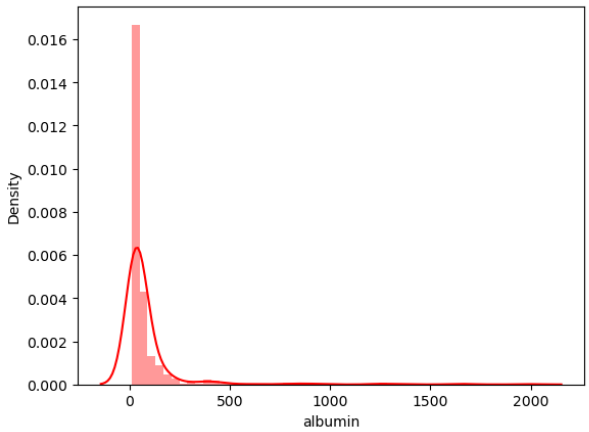
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [34]:
```
sns.distplot(liverdata['albumin'],color='red')
```

Out[34]:
```
<Axes: xlabel='albumin', ylabel='Density'>
```




Type here to search

36°C Mostly sunny ENG 16:01 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

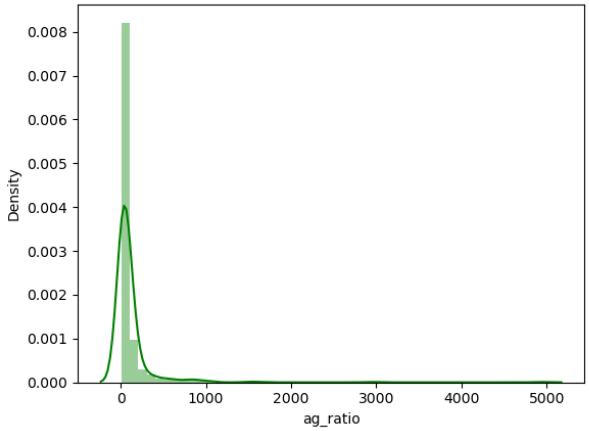
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [35]:
```
sns.distplot(liverdata['ag_ratio'],color='green')
```

Out[35]:
```
<Axes: xlabel='ag_ratio', ylabel='Density'>
```




Type here to search

36°C Mostly sunny ENG 16:02 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

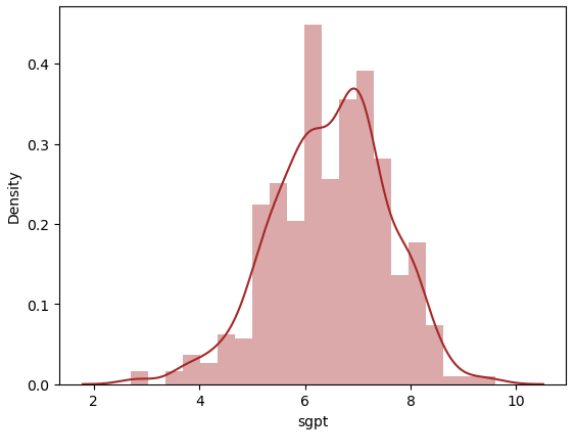
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [36]:
```
sns.distplot(liverdata['sgpt'],color='brown')
```

Out[36]:
```
<Axes: xlabel='sgpt', ylabel='Density'>
```




Type here to search

36°C Mostly sunny ENG 16:04 22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

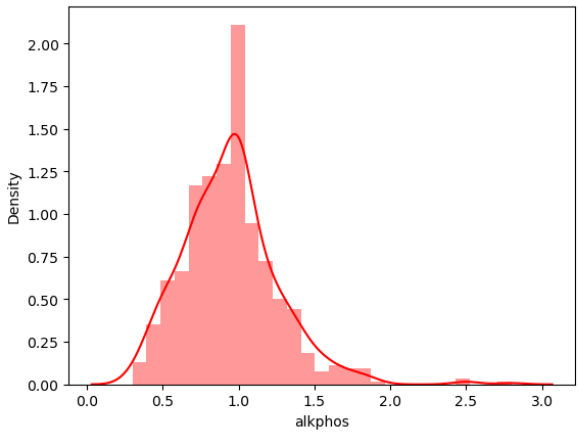
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

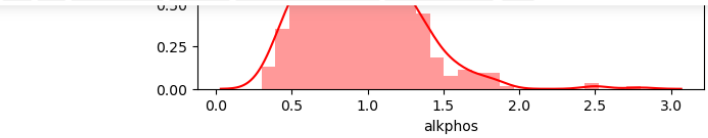
In [38]:
```
sns.distplot(liverdata['alkphos'],color='red')
```

Out[38]:
```
<Axes: xlabel='alkphos', ylabel='Density'>
```




Type here to search

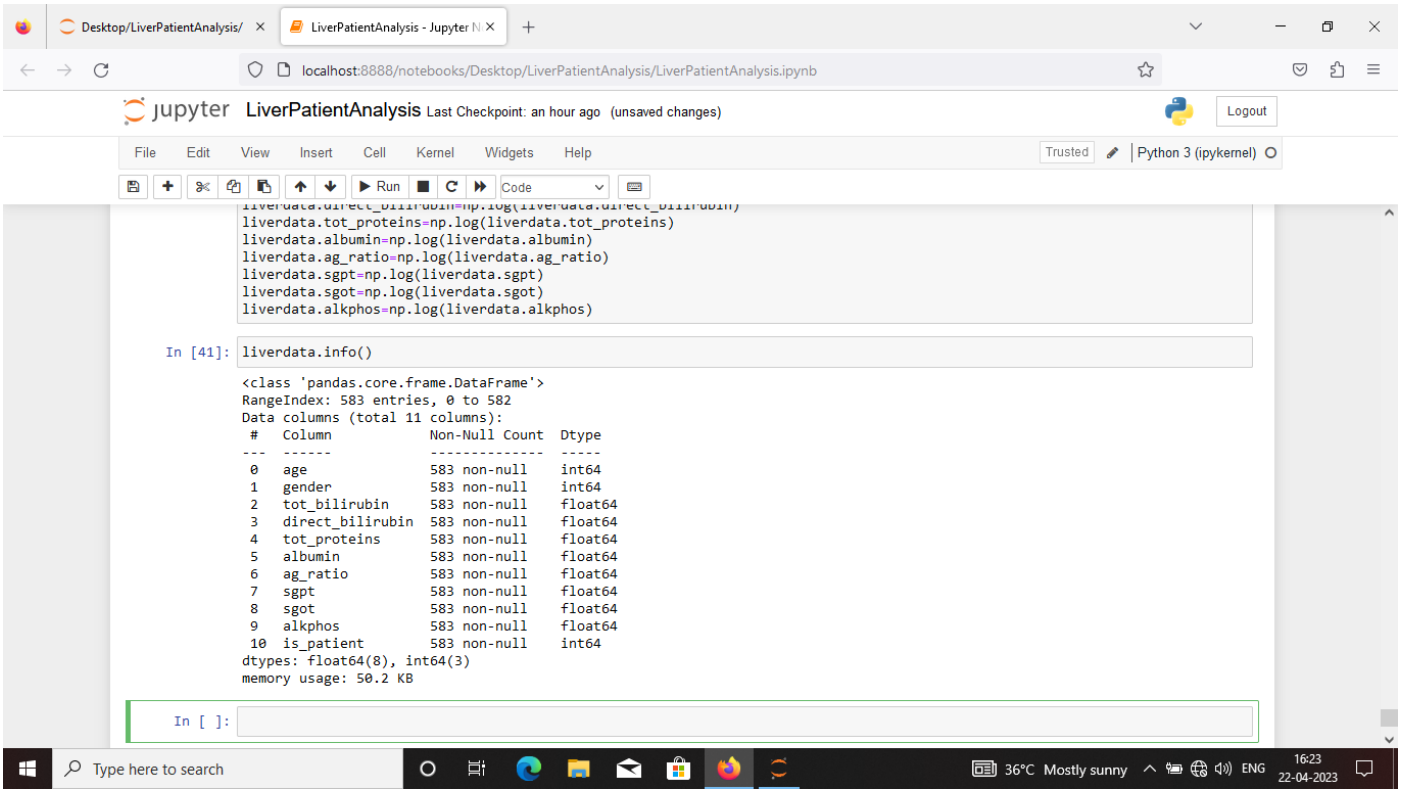
36°C Mostly sunny ENG 16:06 22-04-2023




In [39]:
```
liverdata['tot_bilirubin'].describe()
```

Out[39]:
```
count    583.000000
mean       3.298799
std        6.209522
min        0.400000
25%        0.800000
50%        1.000000
75%        2.600000
max       75.000000
Name: tot_bilirubin, dtype: float64
```

In [ ]:


Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
liverdata.tot_proteins=np.log(liverdata.tot_proteins)
liverdata.albumin=np.log(liverdata.albumin)
liverdata.ag_ratio=np.log(liverdata.ag_ratio)
liverdata.sgpt=np.log(liverdata.sgpt)
liverdata.sgot=np.log(liverdata.sgot)
liverdata.alkphos=np.log(liverdata.alkphos)
In [41]:
liverdata.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 int64
2 tot_bilirubin 583 non-null float64
3 direct_bilirubin 583 non-null float64
4 tot_proteins 583 non-null float64
5 albumin 583 non-null float64
6 ag_ratio 583 non-null float64
7 sgpt 583 non-null float64
8 sgot 583 non-null float64
9 alkphos 583 non-null float64
10 is_patient 583 non-null int64
dtypes: float64(8), int64(3)
memory usage: 50.2 KB
In [ ]:
Type here to search
36°C Mostly sunny
ENG
16:23
22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

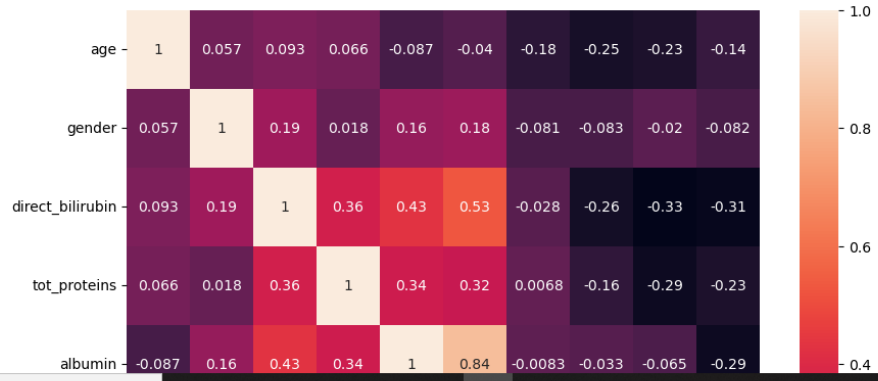
Trusted | Python 3 (ipykernel)

Run | Code

## # Step 6 : Check the correlation

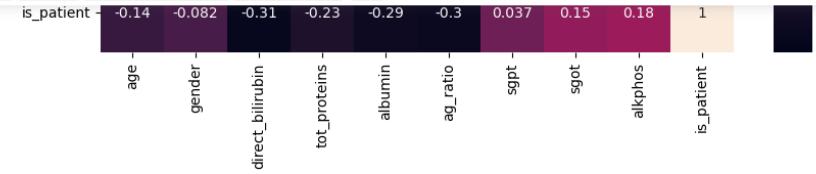
In [47]:
```
corr=liverdata.corr()
plt.figure(figsize=(10,10))
sns.heatmap(data=corr,annot=True)
```

Out[47]: <Axes: >




Type here to search

36°C Mostly sunny ENG 16:30 22-04-2023



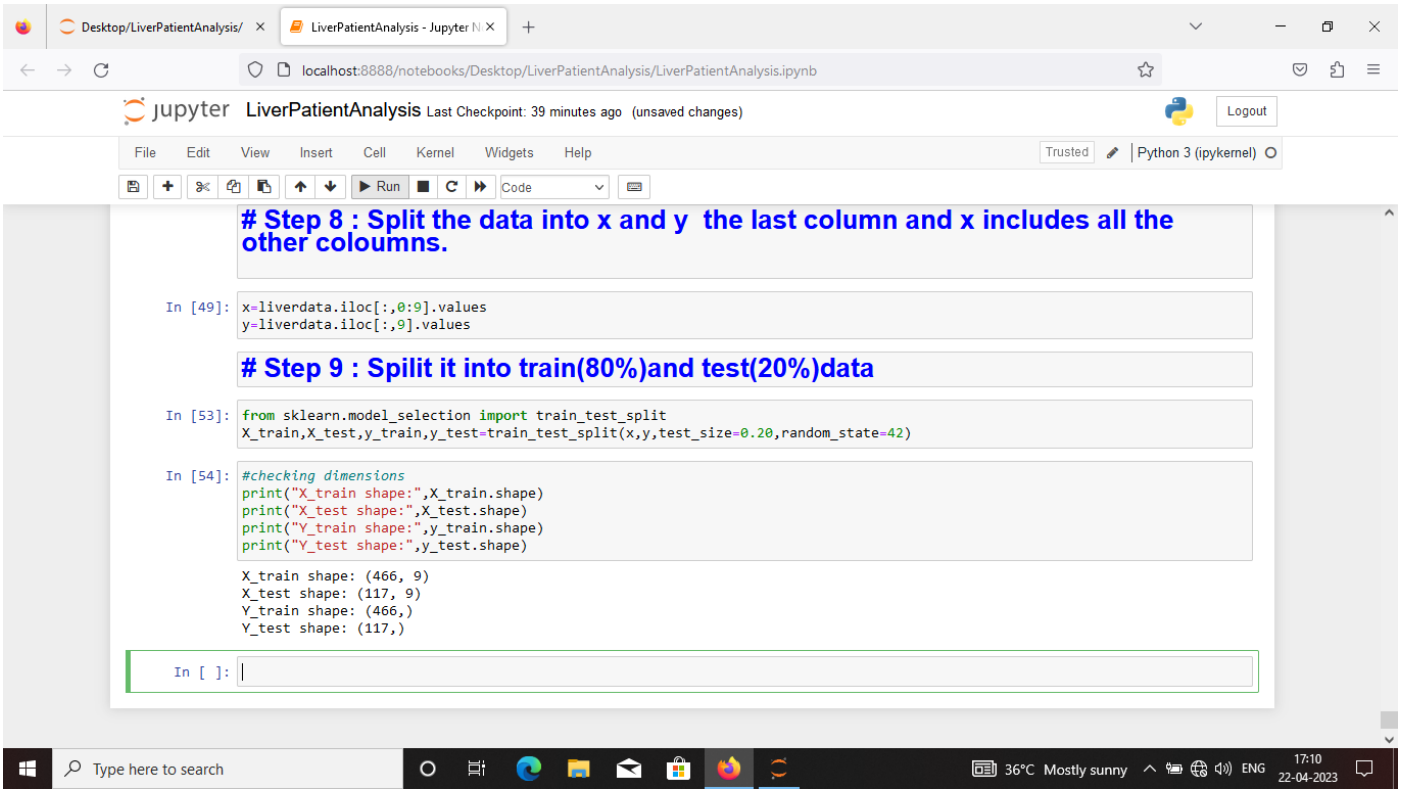

## # Step 7 : Check if any columns contain + or -inf.If so,replace it by nan

In [48]:
```
liverdata.loc[liverdata['alkphos']==-np.inf]
liverdata.loc[liverdata['ag_ratio']==-np.inf]
liverdata.loc[liverdata['albumin']==-np.inf]
liverdata.loc[liverdata['tot_proteins']==-np.inf]
liverdata.loc[liverdata['direct_bilirubin']==-np.inf]
#Check if -ve or +ve infinity present in other coloumns.It is not present.So neednot replace with NaN
```

Out[48]:

| age | gender | direct_bilirubin | tot_proteins | albumin | ag_ratio | sgpt | sgot | alkphos | is_patient |
|---|---|---|---|---|---|---|---|---|---|

In [ ]:


Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: 39 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
# Step 8 : Split the data into x and y the last column and x includes all the other coloumns.
In [49]:
x=liverdata.iloc[:,0:9].values
y=liverdata.iloc[:,9].values
# Step 9 : Spilit it into train(80%)and test(20%)data
In [53]:
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=42)
In [54]:
#checking dimensions
print("X_train shape:",X_train.shape)
print("X_test shape:",X_test.shape)
print("Y_train shape:",y_train.shape)
print("Y_test shape:",y_test.shape)
X_train shape: (466, 9)
X_test shape: (117, 9)
Y_train shape: (466,)
Y_test shape: (117,)
In [ ]:
Type here to search
36°C Mostly sunny
ENG
17:10
22-04-2023

Desktop/LiverPatientAnalysis/ | LiverPatientAnalysis - Jupyter N

localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb

jupyter **LiverPatientAnalysis** Last Checkpoint: 10 hours ago (unsaved changes)

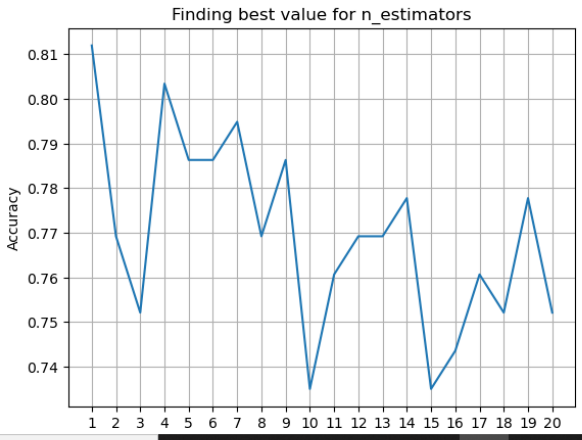
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

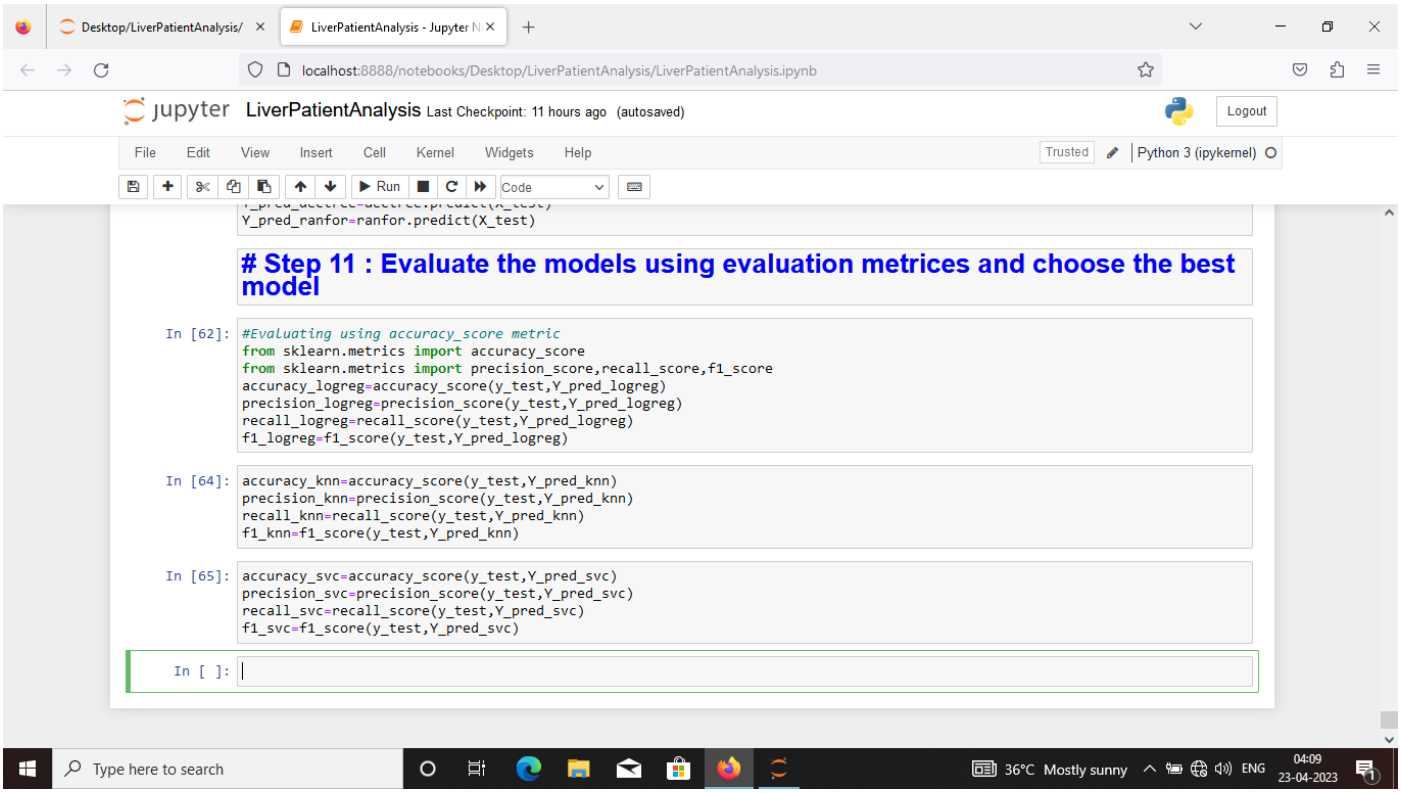
Run | Code

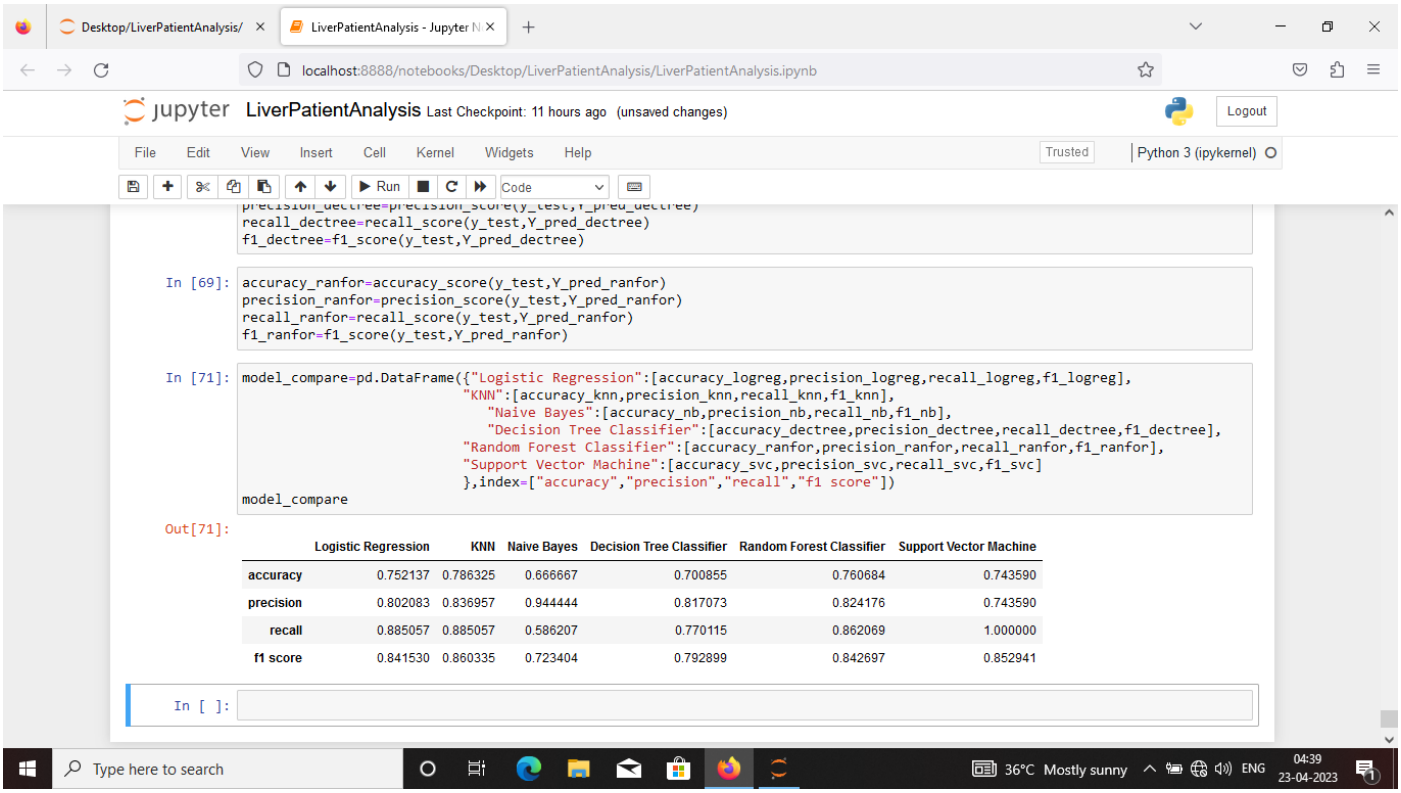
```
plt.xlabel("n_estimators")
plt.ylabel("Accuracy")
plt.grid()
plt.show()
print('High value:',acc.values.max())
```



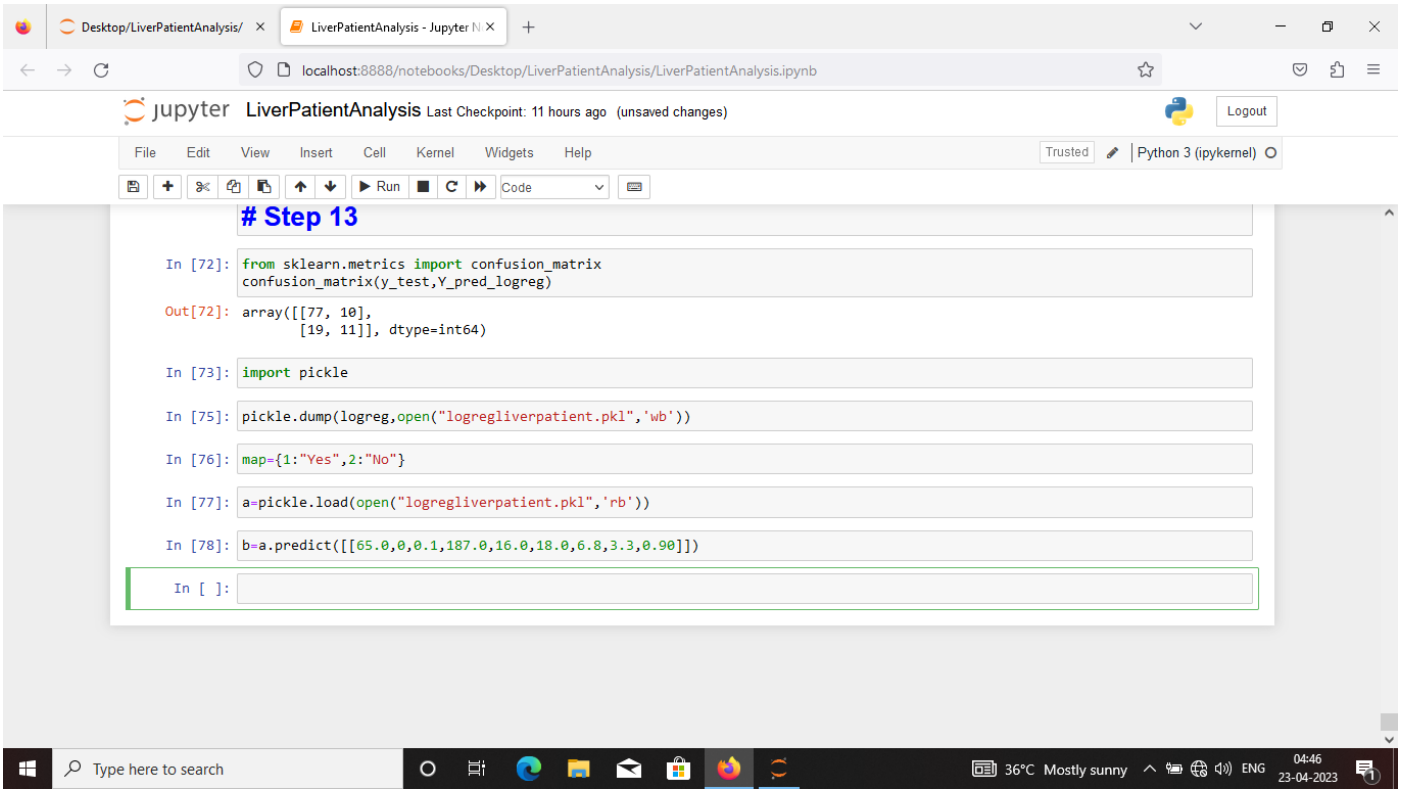

Type here to search

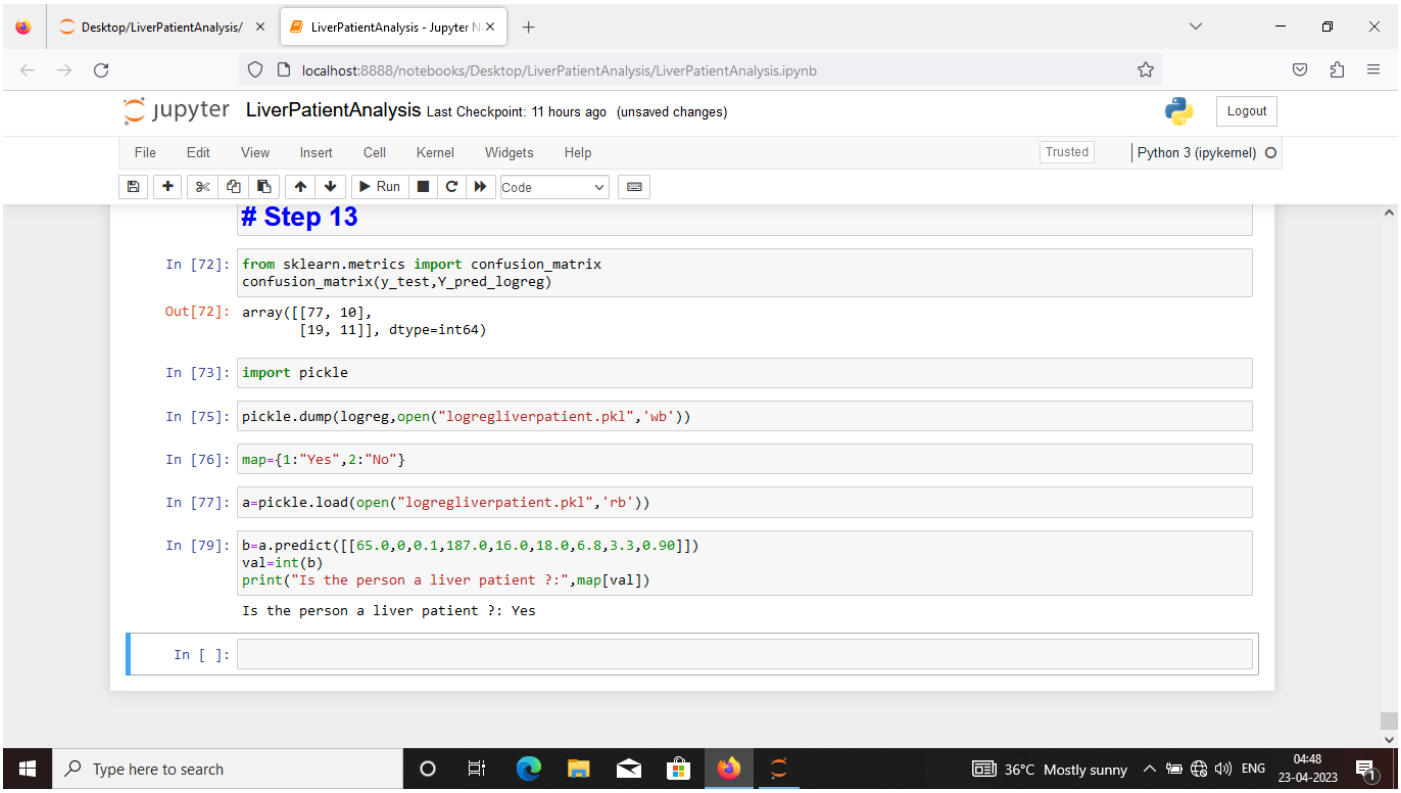
36°C Mostly sunny ENG 03:33 23-04-2023

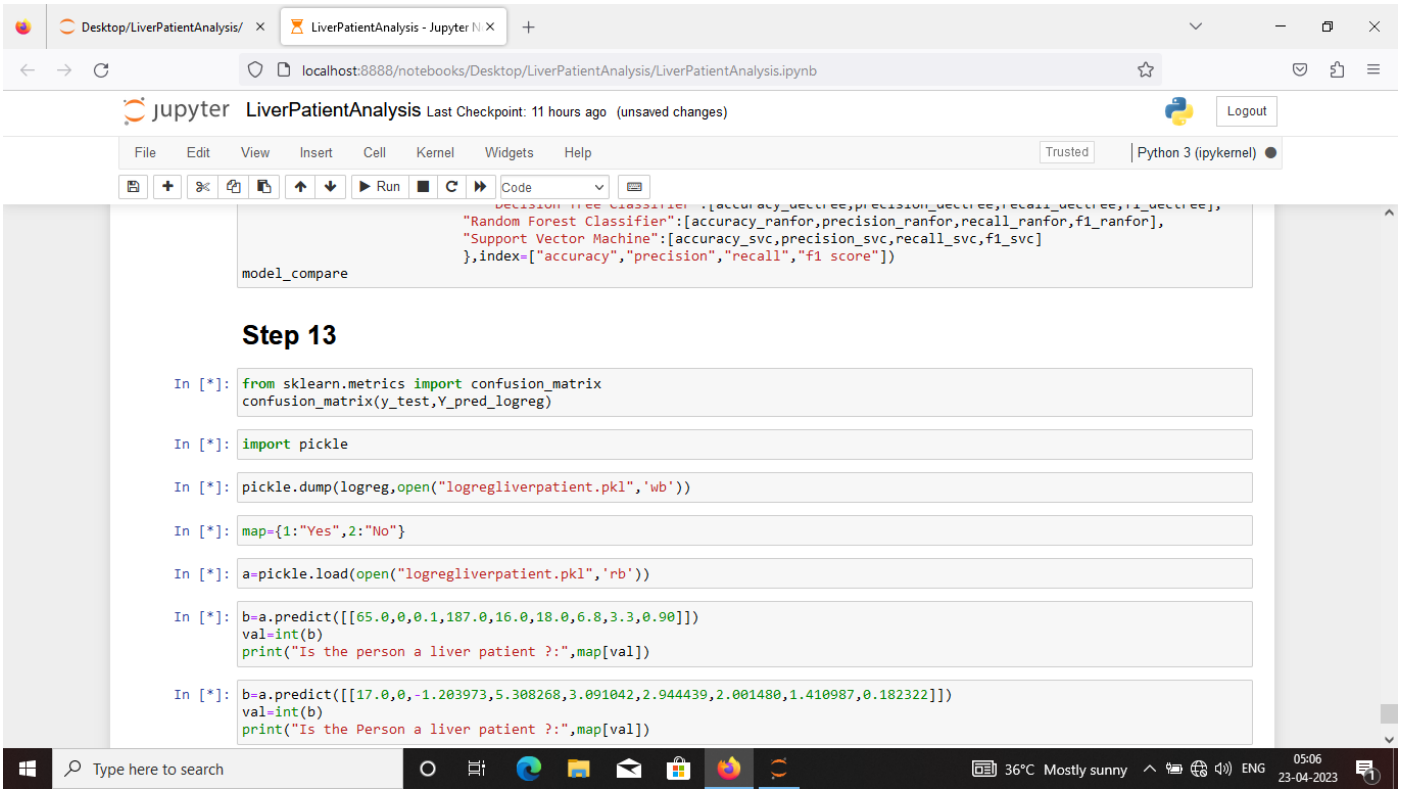

Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: 11 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Y_pred_ranfor=ranfor.predict(X_test)
# Step 11 : Evaluate the models using evaluation metrices and choose the best model
In [62]:
#Evaluating using accuracy_score metric
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score,recall_score,f1_score
accuracy_logreg=accuracy_score(y_test,Y_pred_logreg)
precision_logreg=precision_score(y_test,Y_pred_logreg)
recall_logreg=recall_score(y_test,Y_pred_logreg)
f1_logreg=f1_score(y_test,Y_pred_logreg)
In [64]:
accuracy_knn=accuracy_score(y_test,Y_pred_knn)
precision_knn=precision_score(y_test,Y_pred_knn)
recall_knn=recall_score(y_test,Y_pred_knn)
f1_knn=f1_score(y_test,Y_pred_knn)
In [65]:
accuracy_svc=accuracy_score(y_test,Y_pred_svc)
precision_svc=precision_score(y_test,Y_pred_svc)
recall_svc=recall_score(y_test,Y_pred_svc)
f1_svc=f1_score(y_test,Y_pred_svc)
In [ ]:
Type here to search
36°C Mostly sunny
ENG
04:09
23-04-2023




| | Logistic Regression | KNN | Naive Bayes | Decision Tree Classifier | Random Forest Classifier | Support Vector Machine |
|---|---|---|---|---|---|---|
| accuracy | 0.752137 | 0.786325 | 0.666667 | 0.700855 | 0.760684 | 0.743590 |
| precision | 0.802083 | 0.836957 | 0.944444 | 0.817073 | 0.824176 | 0.743590 |
| recall | 0.885057 | 0.885057 | 0.586207 | 0.770115 | 0.862069 | 1.000000 |
| f1 score | 0.841530 | 0.860335 | 0.723404 | 0.792899 | 0.842697 | 0.852941 |

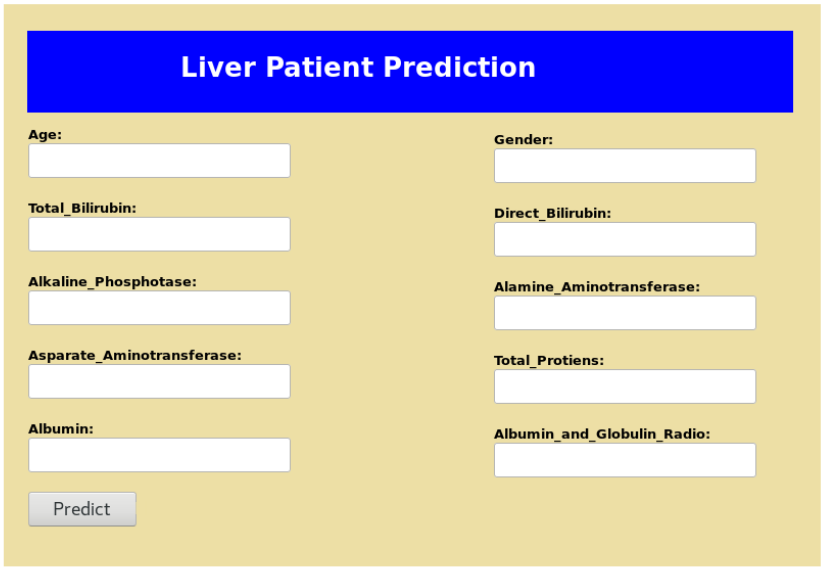

Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: 11 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
# Step 13
In [72]:
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,Y_pred_logreg)
Out[72]:
array([[77, 10],
[19, 11]], dtype=int64)
In [73]:
import pickle
In [75]:
pickle.dump(logreg,open("logregliverpatient.pkl",'wb'))
In [76]:
map={1:"Yes",2:"No"}
In [77]:
a=pickle.load(open("logregliverpatient.pkl",'rb'))
In [78]:
b=a.predict([[65.0,0,0.1,187.0,16.0,18.0,6.8,3.3,0.90]])
In [ ]:
Type here to search
36°C Mostly sunny
ENG
04:46
23-04-2023

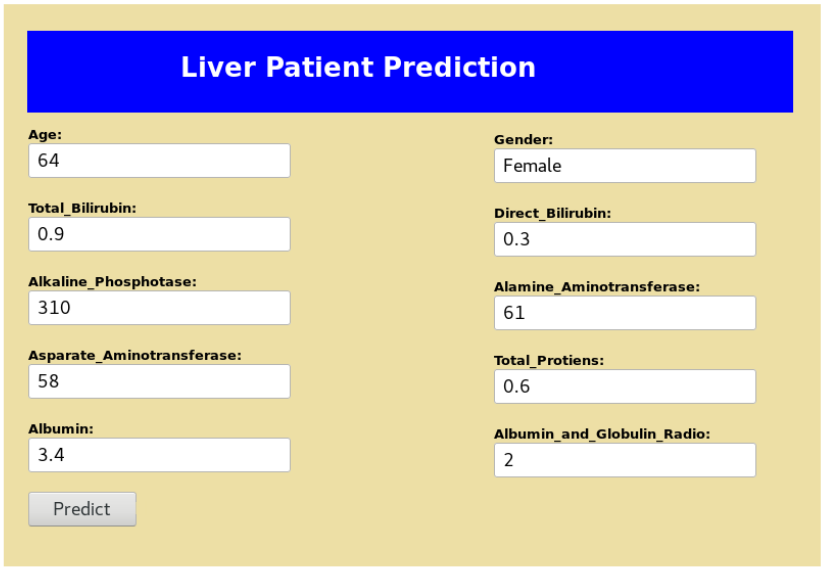

Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: 11 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
# Step 13
In [72]:
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,Y_pred_logreg)
Out[72]:
array([[77, 10],
[19, 11]], dtype=int64)
In [73]:
import pickle
In [75]:
pickle.dump(logreg,open("logregliverpatient.pkl",'wb'))
In [76]:
map={1:"Yes",2:"No"}
In [77]:
a=pickle.load(open("logregliverpatient.pkl",'rb'))
In [79]:
b=a.predict([[65.0,0,0.1,187.0,16.0,18.0,6.8,3.3,0.90]])
val=int(b)
print("Is the person a liver patient ?:",map[val])
Is the person a liver patient ?: Yes
In [ ]:
Type here to search
36°C Mostly sunny
ENG
04:48
23-04-2023


Desktop/LiverPatientAnalysis/
LiverPatientAnalysis - Jupyter N
localhost:8888/notebooks/Desktop/LiverPatientAnalysis/LiverPatientAnalysis.ipynb
jupyter LiverPatientAnalysis Last Checkpoint: 11 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
"Random Forest Classifier":[accuracy_ranfor,precision_ranfor,recall_ranfor,f1_ranfor],
"Support Vector Machine":[accuracy_svc,precision_svc,recall_svc,f1_svc]
},index=["accuracy","precision","recall","f1 score"])
model_compare
Step 13
In [*]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,Y_pred_logreg)
In [*]: import pickle
In [*]: pickle.dump(logreg,open("logregliverpatient.pkl",'wb'))
In [*]: map={1:"Yes",2:"No"}
In [*]: a=pickle.load(open("logregliverpatient.pkl",'rb'))
In [*]: b=a.predict([[65.0,0,0.1,187.0,16.0,18.0,6.8,3.3,0.90]])
val=int(b)
print("Is the person a liver patient ?:",map[val])
In [*]: b=a.predict([[17.0,0,-1.203973,5.308268,3.091042,2.944439,2.001480,1.410987,0.182322]])
val=int(b)
print("Is the Person a liver patient ?:",map[val])
Type here to search
36°C Mostly sunny
ENG
05:06
23-04-2023

## Introduction

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumtion liver disease arises.Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctor diagnose the disease within a short duration of tim. Descovering the existence of liver disease at an early stage is a complex task for the doctor. The main objective of this paper is to analyse the parameters of various clasdification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This paper focuses on the related works of various authirs on liver disease such that algorithms were implemented using Weka also evaluated. This paper comparea various classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm with an aim to identify the best technique.Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver dieaserecommended


Liver Patient Prediction
Age:
Gender:
Total_Bilirubin:
Direct_Bilirubin:
Alkaline_Phosphotase:
Alamine_Aminotransferase:
Asparate_Aminotransferase:
Total_Protiens:
Albumin:
Albumin_and_Globulin_Radio:
Predict


Liver Patient Prediction
Age:
64
Gender:
Female
Total_Bilirubin:
0.9
Direct_Bilirubin:
0.3
Alkaline_Phosphotase:
310
Alamine_Aminotransferase:
61
Asparate_Aminotransferase:
58
Total_Protiens:
0.6
Albumin:
3.4
Albumin_and_Globulin_Radio:
2
Predict

**3. RESULT :**

Liver Patient Analysis

Home

## Introduction

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumtion liver disease arises.Early pre
disease using classification algorithms is an efficacious task that can help the doctor diagnose the disease within a short duration of tim
existence of liver disease at an early stage is a complex task for the doctor. The main objective of this paper is to analyse the parameter
clasdification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. Th
the related works of various authirs on liver disease such that algorithms were implemented using Weka also evaluated. This paper com
classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm with an aim to identify the best techniqu
study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver

## Liver Patient Prediction

**Age:**

**Gender:**

**Total_Bilirubin:**

**Direct_Bilirubin:**

**Alkaline_Phosphotase:**

**Alamine_Aminotransferase:**

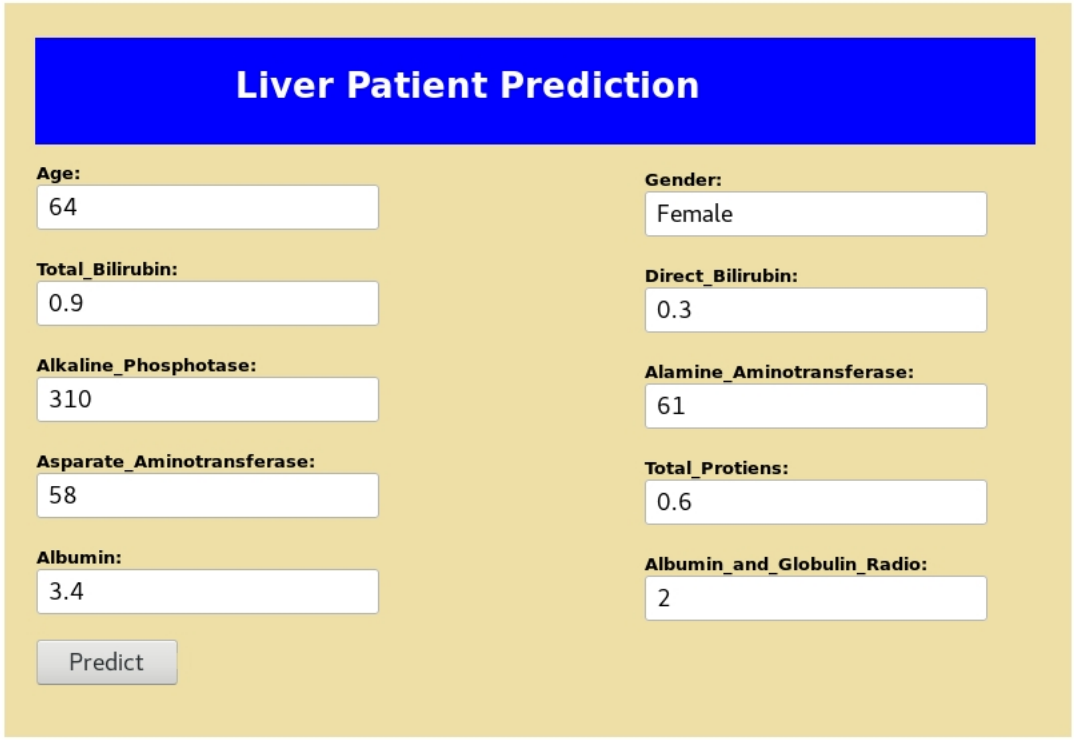
**Asparate_Aminotransferase:**

**Total_Protiens:**

**Albumin:**

**Albumin_and_Globulin_Radio:**

Predict


Liver Patient Prediction
Age:
64
Gender:
Female
Total_Bilirubin:
0.9
Direct_Bilirubin:
0.3
Alkaline_Phosphotase:
310
Alamine_Aminotransferase:
61
Asparate_Aminotransferase:
58
Total_Protiens:
0.6
Albumin:
3.4
Albumin_and_Globulin_Radio:
2
Predict

# Liver Patient Prediction

**You have a liver desease problem, You must and should consult a doctor. Take care**