# Indian Liver Patient Data Modelling and Analysis:

## INTRODUCTION:

a. **Overview:**

Liver is the largest organ in the abdomen. This is the primary organ for maintaining the chemicals like glucose, balancing so many nutrients, fat, vitamins, cholesterol and hormones. Liver disease prevents normal liver function. Mainly due to the large amount of alcohol consumption liver disease develops. Early detection of liver disease using classification algorithms is an effective activity that can help doctors diagnose the disease in a short period of time. Early detection of liver disease is a daunting task for doctors. The main purpose of our project is to analyze the parameters of the various classification algorithms and compare their predictive accuracy to determine the best stage for determining liver disease.

b. **Purpose:**

The use of this project is to focus on the related activities of various authors on liver disease so that the algorithms are implemented using the tool which is a typical machine learning software. Various traits important in liver disease prognosis are assessed and a database of liver patients is also evaluated. This project compares various classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm along with decision tree algorithm and KNN algorithm for the purpose of identifying the best method.

## LITERATURE SURVEY:

a. **Existing problem:**

**Rong-Ho Lin** [9] proposed to predict the accuracy of liver disease using case-based reasoning (CBR) and classification and regression tree (CART) approach. He also integrates CART and CBR for the diagnosis of liver diseases. This model included two major steps. (1) CART To diagnose whether a patient suffers from liver disease using CART. (2) To predict which types of Liver disease affected for patients using CBR.He also [18], proposed to determine whether patients suffer from liver disease or not using case-based

reasoning, artificial neural networks and analytic hierarchy methods. They also predict which types of liver disease the human body.

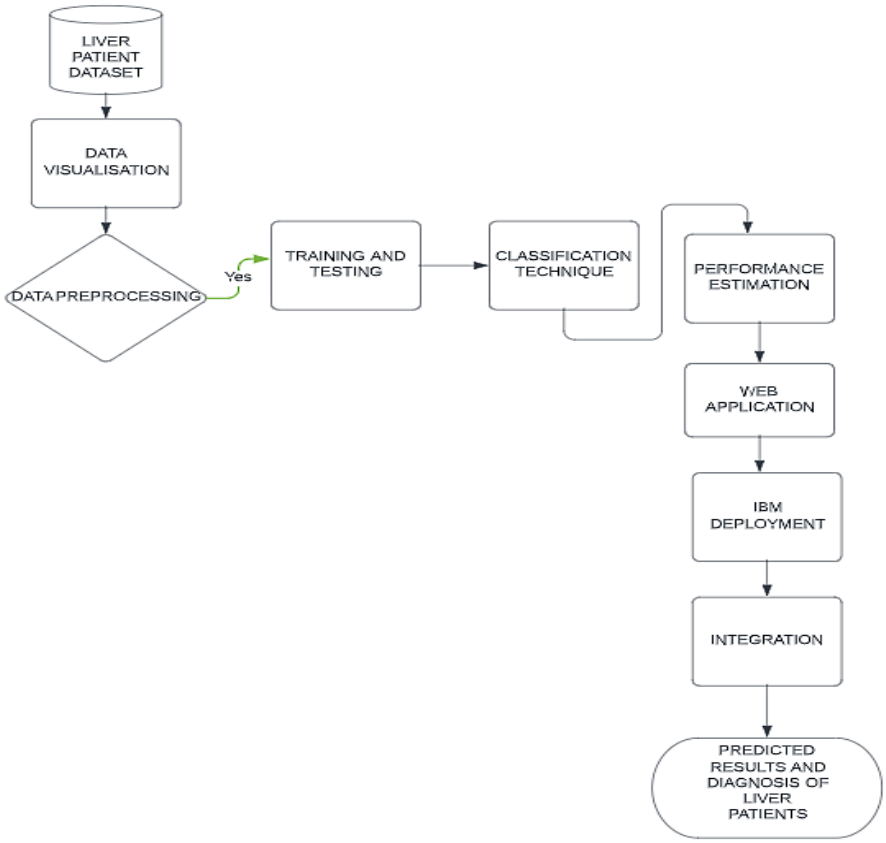
**Kemal Polat** et al.,[22] proposed the resource allocation mechanism of AIRS was changed with a new one decided by Fuzzy-Logic. This approach called Fuzzy- AIRS was used as a classifier in the diagnosis of Liver Disorders. In this Classification, accuracies were evaluated by comparing them with reported classifier's accuracy, time and number of resources.

b. **Proposed solution:**

We propose to use various Machine Learning Algorithms to build a data model based on the available dataset. Our solution aims to predict whether a patient has liver Disease or not based on various parameters derived from the dataset. We aim to deploy this model in the cloud, and integrate it with a web application for further usage.

# THEORETICAL ANALYSIS

### 3.1. Block Diagram:



### 3.2. Hardware / Software designing:

We are using **Jupyter Notebook** as the main software.

**The required libraries to be imported to Python script are:**

**Numpy**: It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.

**Pandas**: It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

**Matplotlib**: Visualisation with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python.

**Seaborn**: Seaborn is a library for making statistical graphics in Python. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

**Pickle**: The pickle module implements serialization protocol, which provides an ability to save and later load Python objects using special binary format.

**Scikit-learn**: Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. We can also call sklearn as short.

**Sklearn.model_selection**: train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

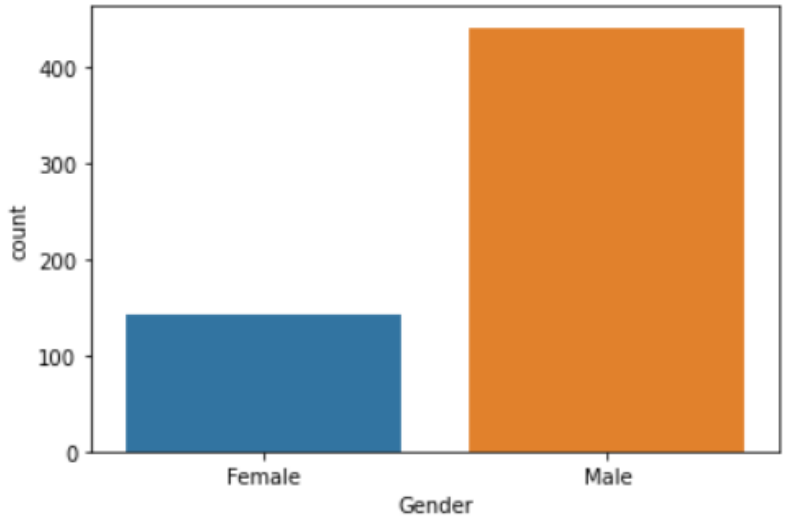
**Test_size():** This parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset. If you're specifying this parameter, you can ignore the next parameter.

We used **Flask** to create the web application and HTML to create the front end part.

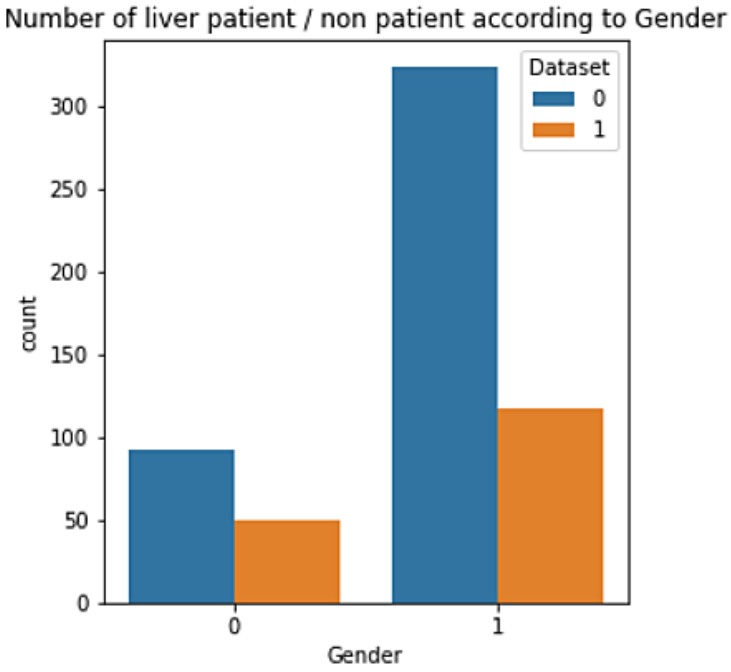
# EXPERIMENTAL INVESTIGATIONS:

**a. Plotting the number of male and female patients:**

```
Number of female patients:  142
Number of male patients:  441
'0' represents the number of female patients.
'1' represents the number of male patients.
```

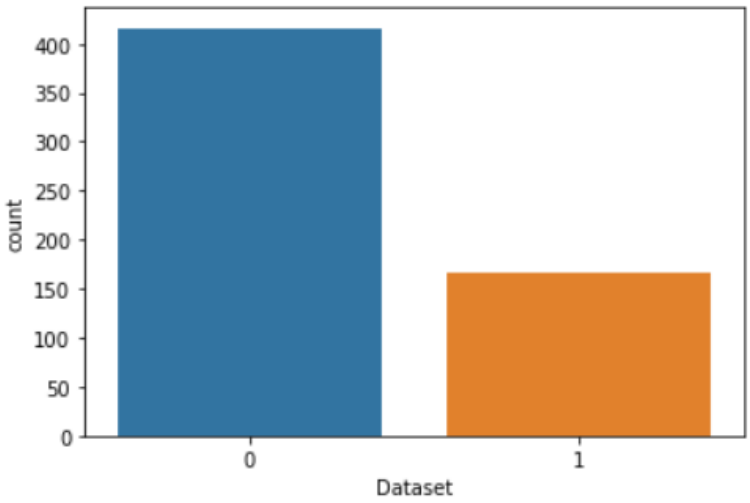



**b. Plotting the number of patient / nonpatient according to gender:**

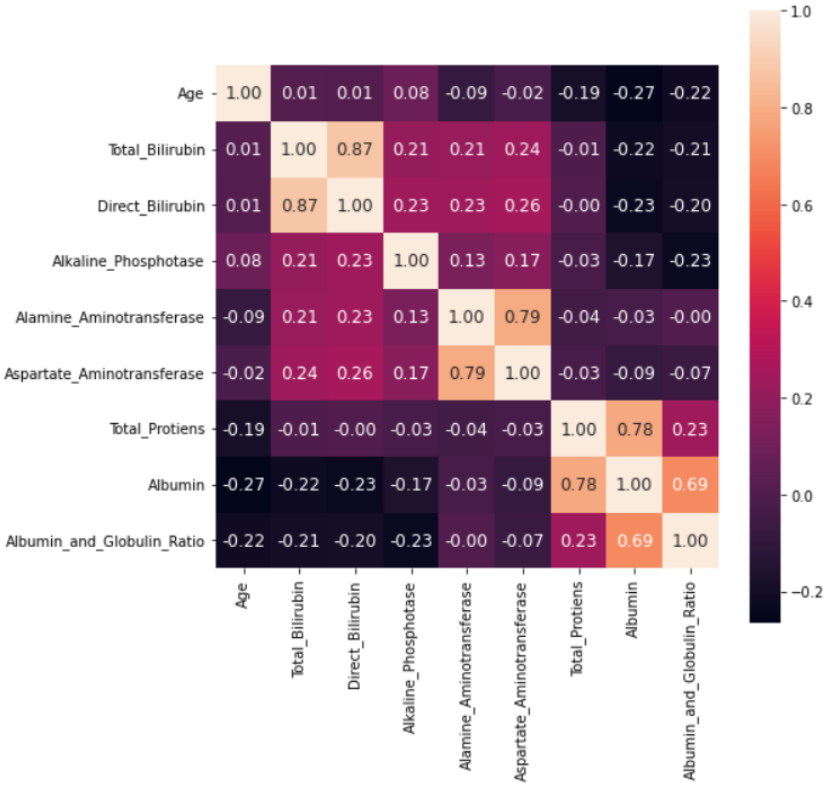


**c. Plotting the number of people with liver disease vs the number of people who doesn't have liver disease:**

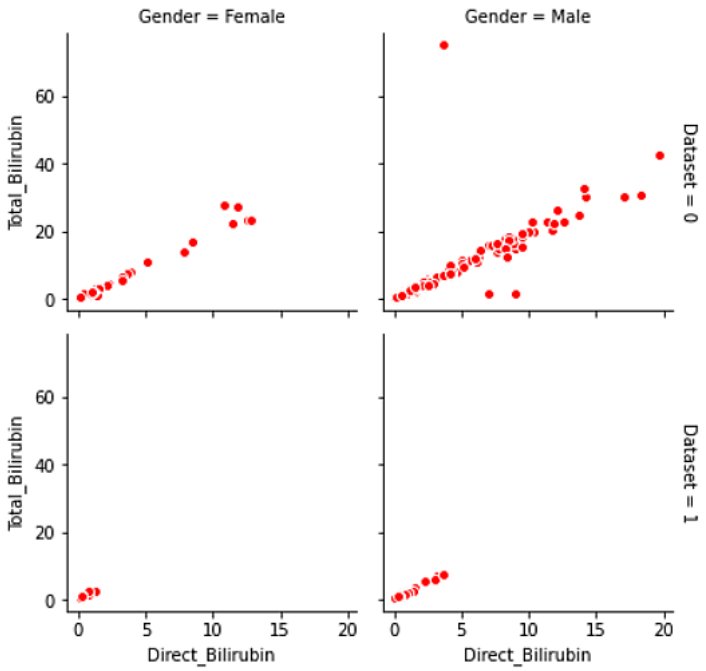
```
Number of patients with liver disease:  416
Number of patients without liver disease:  167
'0' represents the number of patients with liver disease.
'1' represents the number of patients who doesnt have liver disease.
```



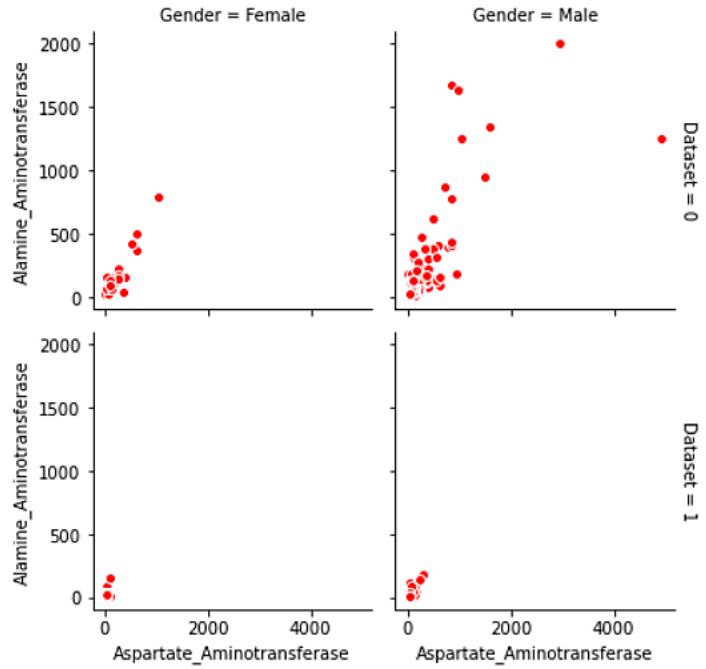

**d. Heat map of the correlation factor b/w each column:**



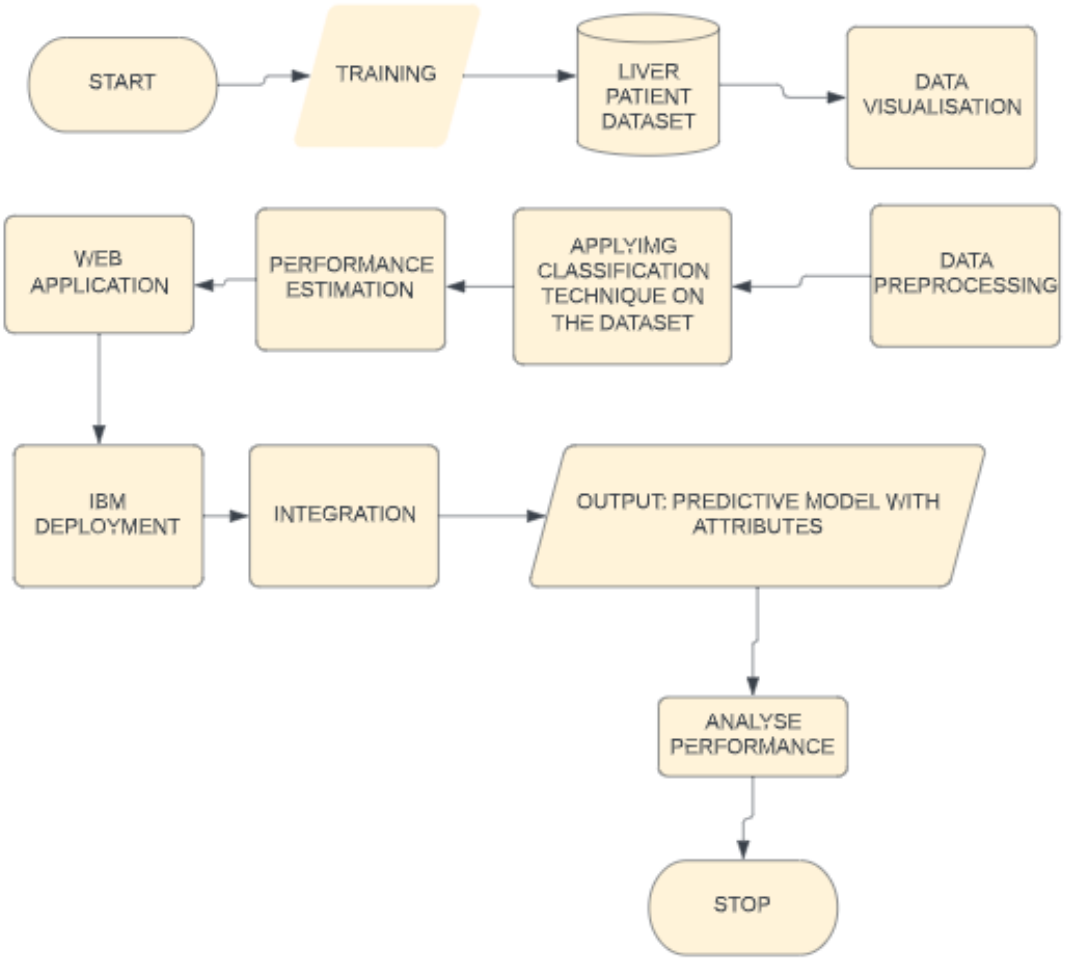
**e. Plotting Gender(Male/Female) along with Total Bilirubin and Direct Bilirubin:**




**f. Plotting Gender(Male/Female) along with Aspartate Aminotransferase, Alanine Aminotransferase:**



# FLOWCHART




# RESULT:

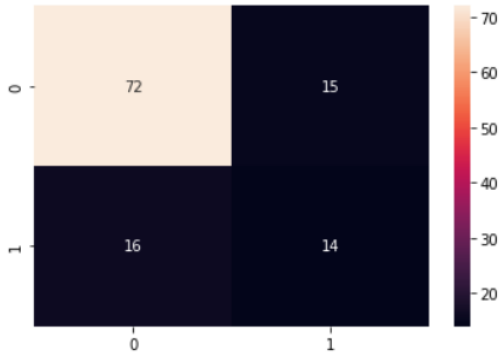
We have applied 6 different Classifiacation Agorithms on the dataset and have found the following result . However we have displayed the resulsts of only the 4 best performing algorithms .

1. **RANDOM FOREST CLASSIFICATION :**

   According to our analysis Random Forest Classification was the best performing algorithm with 73.5 % accuracy and an AUC score of 0.65.

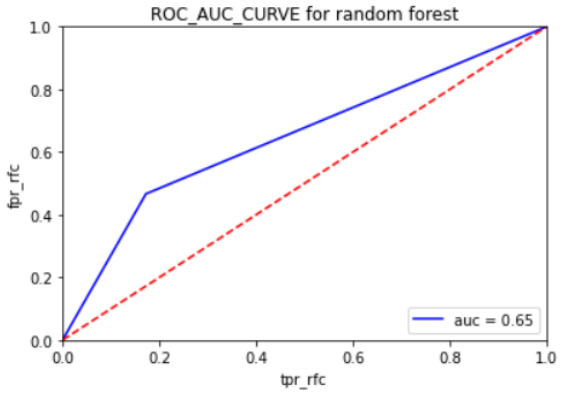
**a. Heat Map of the Confusion Matrix of the Random Forest Classification Algorithm:**

```
In [210]:
1 cm_rfc = confusion_matrix(y_test, y_pred_rfc)
2 cm_rfc

Out[210]: array([[72, 15],
                 [16, 14]], dtype=int64)

In [211]:
1 sns.heatmap(cm_rfc, annot = True)

Out[211]: <AxesSubplot:>
```




**b. The image below shows the ROC_AUC Curve of the Random Forest Classification Algorithm model trained on the dataset:**

```
In [197]:
1 plt.title("ROC_AUC_CURVE for random forest")
2 plt.plot(fpr_rfc,tpr_rfc, 'b', label = 'auc = %0.2f'%roc_auc)
3 plt.legend(loc = 'lower right')
4 plt.plot([0,1],[0,1], 'r--')
5 plt.xlim([0,1])
6 plt.ylim([0,1])
7 plt.xlabel('tpr_rfc')
8 plt.ylabel('fpr_rfc')

Out[197]: Text(0, 0.5, 'fpr_rfc')
```

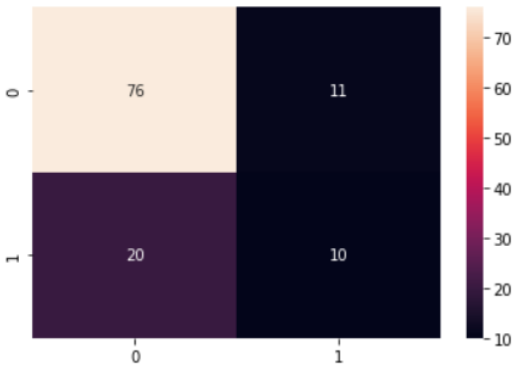


## 2. Logistic Regression:

Upon applying the logistic regression algorithm on this dataset we found an accuracy of 73.5% with an AUC score of 0.60.

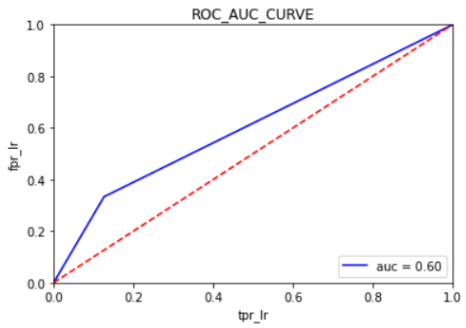
**Heatmap of the confusion matrix of the Logistic Regression Algorithm:**

```
In [132]:
cm_lr = confusion_matrix(y_test, y_pred_lr)
cm_lr

Out[132]:
array([[76, 11],
       [20, 10]], dtype=int64)

In [133]:
import seaborn as sns
sns.heatmap(cm_lr,annot=True)

Out[133]:
<AxesSubplot:>
```




**The image represents the ROC_AUC Curve of the Logistic Regression Model on the dataset:**

```
In [138]:
import matplotlib.pyplot as plt
plt.title("ROC_AUC_CURVE")
plt.plot(fpr_lr,tpr_lr, 'b', label = 'auc = %0.2f'%roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0,1],[0,1], 'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('tpr_lr')
plt.ylabel('fpr_lr')

Out[138]:
Text(0, 0.5, 'fpr_lr')
```

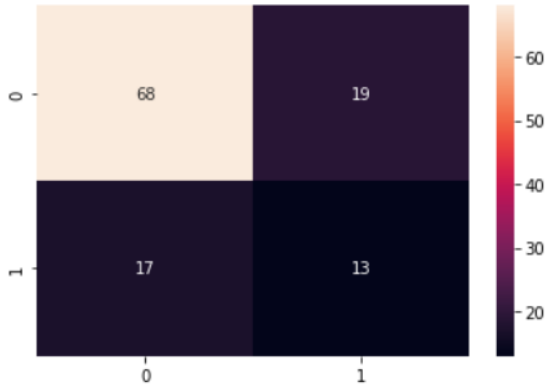


## 3. K Nearest Neighbor:

Upon performing the KNN algorithm on the dataset , we had an accuracy of 69.2% and an AUC score of 0.61.

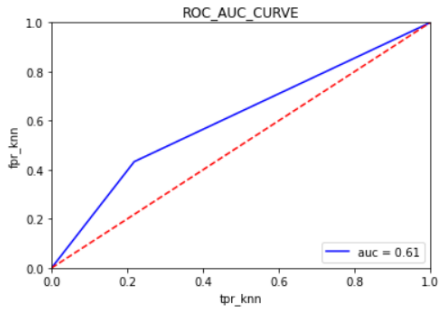
**Heat Map of the Confusion Matrix of KNN algorithms:**

```
In [144]:
1 cm_knn = confusion_matrix(y_test, y_pred_knn)
2 cm_knn

Out[144]: array([[68, 19],
                 [17, 13]], dtype=int64)

In [145]:
1 sns.heatmap(cm_knn,annot=True)

Out[145]: <AxesSubplot:>
```




**ROC_AUC Curve of the KNN Algorithm applied on the dataset:**

```
In [150]:
1 import matplotlib.pyplot as plt
2 plt.title("ROC_AUC_CURVE")
3 plt.plot(fpr_knn,tpr_knn, 'b', label = 'auc = %0.2f'%roc_auc)
4 plt.legend(loc = 'lower right')
5 plt.plot([0,1],[0,1], 'r--')
6 plt.xlim([0,1])
7 plt.ylim([0,1])
8 plt.xlabel('tpr_knn')
9 plt.ylabel('fpr_knn')

Out[150]: Text(0, 0.5, 'fpr_knn')
```

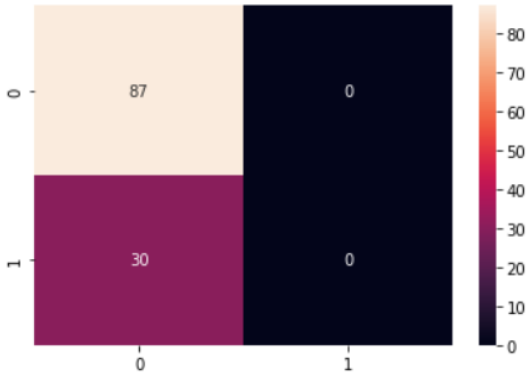


## 4. Support Vector Machine :

Upon applying the Support Vector Machine Algorithm on the dataset we find an accuracy of 73.5% and AUC score of 0.5 .

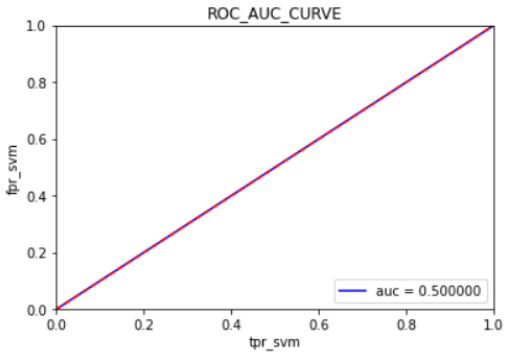
### Heat Map of the Confusion Matrix of SVM :

```
In [156]:
1 cm_svm = confusion_matrix(y_test,y_pred_svm)
2 cm_svm

Out[156]: array([[87,  0],
                 [30,  0]], dtype=int64)

In [157]:
1 sns.heatmap(cm_svm,annot=True)

Out[157]: <AxesSubplot:>
```




### ROC_AUC Curve of the SVM :

```
In [162]:
1 import matplotlib.pyplot as plt
2 plt.title("ROC_AUC_CURVE")
3 plt.plot(fpr_svm,tpr_svm, 'b', label = 'auc = %f'%roc_auc)
4 plt.legend(loc = 'lower right')
5 plt.plot([0,1],[0,1], 'r--')
6 plt.xlim([0,1])
7 plt.ylim([0,1])
8 plt.xlabel('tpr_svm')
9 plt.ylabel('fpr_svm')

Out[162]: Text(0, 0.5, 'fpr_svm')
```



# ADVANTAGES & DISADVANTAGES

**List of advantages and disadvantages of the proposed solution:**

**a. Advantages:**

1. It can be widely used in Medical Field to predict liver disease.
2. It would reduce the burden on doctors and medical staff leading to a more accurate treatment of the patients.

**b. Disadvantages:**

1. The cost of misclassification of a single patient can be very high.
2. Even the best performing Machine Learning Algorithm may not be 100% correct hence there will always be a risk of error.

# APPLICATIONS

**The areas where this solution can be applied:**

i) Medical industry for predicting the disease.

ii) Analyzing medical information with the images.

iii) Analyzing the enzymesin's levels which will lead to diagnosing liver diseases from the blood.

# CONCLUSION

The main objective of this study was to provide a summary of the classification algorithms in the field of data-driven predictive data for liver disease. In this study, various classification algorithms were analyzed to help doctors predict liver disease early. The purpose of this study was achieved by conducting comparative research in various papers. Based on this study, Random Forest has a much higher accuracy than other algorithms and can be used continuously in predicting user-recommended liver disease.

# FUTURE SCOPE

**Our project can be applied in the future in the following ways:**

a. In the upcoming future, we will convert our web application to an android application with more enhanced and innovative features.

b. We will also integrate digital marketing so that different hospitals and nursing homes can use our application.

# BIBLIOGRAPHY

## References:


1. https://www.researchgate.net/publication/327838226_Classification_of_Liver_Patient_Dataset_Using_Machine_Learning_Algorithms
2. https://www.hindawi.com/journals/jhe/2020/6680002/
3. https://core.ac.uk/download/pdf/231162636.pdf
4. https://www.irjet.net/archives/V5/i1/IRJET-V5I142.pdf


## Appendix:

### Google Drive Link to access the project files and source code:

https://drive.google.com/drive/folders/1JeSivngZQOsb1Mw2WQiP0eF7EsiR8zLc?usp=sharing