A Project Report on

LIVER CANCER PREDICTION USING MACHINE LEARNING

Project Report submitted in partial Fulfilment of the requirement for the award of

Machine Learning using Python (CA3211)

SUBMITTED BY:-

BISWAJIT BISWAL, 2061020070
BANASMITA SUBHADARSHINI, 2061020077

DEPARTMENT OF MCA (2020-22)

ITER, SOA UNIVERSITY

Jagamohan Nagar, Jagamara, Bhubaneswar, Odisha – 75103

Problem Statement:-

One of the most complex internal biological structures in the human body is liver .Upper right hand part of the abdomen is located by the liver which is reddish brown in colour and measures eight and half inches . Liver is wedge shaped gland normally weighs 1440grams to 1660 grams. Liver is divided into Left lobe and right lobe and filters 1.5L of blood per minute approximately. Liver functions include production of bile, production of cholesterol and special proteins, metabolizes drug and detoxifies chemicals. Liver is also having some diseases such as hepatitis, cancer, cirrhosis, hemochromatosis and jaundice. Liver cancer is the most dangerous cancer among variety of cancer. Due to this every third living is cause of death and which is nearly a sixth most common cancer in the world. Liver cancer is also known by the name hepatic cancer and most of the liver cancer is common to Hepatic cellular carcinoma (HCC). Liver cancer is the uncontrolled growing of tissue within the liver. Tumours are of two types such as non-cancerous cells (benign) and cancerous cells (malignant). There are 12000 deaths per year in world due to liver cancer. To avoid this, problem need to be analyzed in earlier stages because earlier detection can help doctors to save lives and does not make very much complication on the human health**.** The aim of this paper is to extend the better comprehension of different Machine Learning Techniques (MLT) used in liver lesion detection and to identify the important research orientation in image processing. With the development of machine learning (ML) algorithms, a growing number of predictive models have been established for predicting the therapeutic outcome of patients with hepatocellular carcinoma (HCC) after various treatment modalities. By using the different combinations of clinical and radiological variables, ML algorithms can simulate human learning to detect hidden patterns within the data and play a critical role in artificial intelligence techniques. Compared to traditional statistical methods, ML methods have greater predictive effects. ML algorithms are widely applied in nearly all steps of model establishment, such as imaging feature extraction, predictive factor classification, and model development. Therefore, this review presents the literature pertaining to ML algorithms and aims to summarize the strengths and limitations of ML, as well as its potential value in prognostic prediction, after various treatment modalities for HCC.

Introduction:-

Hepatocellular carcinoma (HCC) is an aggressive tumour which remains the second-most frequent cause of cancer death worldwide. According to the different statuses of patients with HCC, several guidelines recommend various treatment strategies. Due to the aggressive biological behavior of HCC, recurrence is not uncommon. Therefore, it is essential to predict therapeutic outcomes prior to treatment so that physicians can design a personalized therapeutic strategy for each patient. The conventional process of model establishment is selecting the appropriate predictors, utilizing them for statistical analysis and ultimately deriving a multivariate predictive model. However, predictive models developed by traditional statistical methods, such as the logistic regression (LR) model and Cox proportional hazards model, are not reliable because the factors included in the models are too simple and utilize a low evidence level. Machine learning (ML) is a powerful tool for generating high-level medical features or combining quantitative radionics parameters with efficient algorithms .ML algorithms simulate human learning to detect hidden patterns within HCC therapeutic data that are clearer than those derived from traditional statistical methods. With this in mind, ML algorithm has been used in many studies to predict the therapeutic outcome of HCC patients. Thus, in this review, the advantages and disadvantages of each ML algorithm are clarified, and relevant literature on the prediction of therapeutic outcomes after various treatment modalities for HCC is described. Liver cancer is the most dangerous cancer among variety of cancer. Due to this every third living is cause of death and which is nearly a sixth most common cancer in the world. Liver cancer is also known by the name hepatic cancer and most of the liver cancer is common to Hepatic cellular carcinoma (HCC). Liver cancer is the uncontrolled growing of tissue within the liver. Tumours are of two types such as non-cancerous cells (benign) and cancerous cells (malignant). There are 12000 deaths per year in world due to liver cancer. To avoid this, problem need to be analysed in earlier stages because earlier detection can help doctors to save lives and does not make very much complication on the human health. There are various techniques to acquire the image of liver from the patients those are Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and Ultra Sounds but CT image is represented as accurate liver cancer diagnosis imaging modularity. Hence Computed Tomography has extent use in the field of medical technology. However detection of the liver lesion, liver image segmentation and Liver lesion extraction are crucial because it requires experienced radiologist to identify differentiable tissues between liver and non-liver. Sometimes experienced radiologist also failed to identify the tumour in earlier stages because of tumour which are invisible to the human eye. Generally there are many improvements in field of medical imaging techniques such as image processing machine learning techniques and artificial intelligence and these technologies can be used by experience radiologist. Together with experienced radiologist and medical technology for computer aided diagnosis results in the accurate characterization of liver lesion. These techniques will provide clinical assistance to the doctors to improve the diagnosis and maximizes the accuracy of the diagnosis. This technique helps in avoiding surgery and biopsy risks toward the victim. Tumour extractions in the liver CT images are absolutely necessary process in computer aided surgery and computer aided nature of illness identification. But still authoritative analysis and prior detection of liver cancer is a significant difficulty in the field of practical radiology doctors should know the feature of the tumour in order to give effective treatment for victim also helps doctors in further diagnosis. Any general method of automatic/semi-automatic computer aided system will help doctors to provide the effective treatment for the patients by diagnosing the liver cancer feature. Liver cancer, also known as hepatic cancer is a cancer which starts in the liver, and not from another organ which eventually migrates to the liver. In other words, there may be cancers which start from somewhere else and end up in the liver - those are not (primary) liver cancers. Cancers that originate in the liver are known as primary liver cancers. Liver cancer consists of malignant hepatic tumours (growths) in or on the liver. The most common type of liver cancer is hepatocellular carcinoma (or hematoma or HCC), and it tends to affect males more than females. According to the National Health Service (NHS), UK, approximately 1,500 people in the United Kingdom die from HCC each year. The World Health Organization (WHO says that liver cancer as a cause of death is reported at less than 30 cases per 100,000 people worldwide, with rates in parts of Africa and Eastern Asia being particularly high. Experts say that common causes of HCC are regular high alcohol consumption, having unprotected sex and injecting drugs with shared needles. Signs and symptoms of liver cancer tend not to be felt or noticed until the cancer is well advanced. Hepatocellular carcinoma (HCC) signs and symptoms may include Jaundice, Abdominal pain, Unexplained weight loss, Hepatomegaly, Fatigue, Nausea, Emesis (vomiting), Back pain, General itching, Fever. Liver cancer, if not diagnosed early is much more difficult to get rid of. The only way to know whether you have liver cancer early on is through screening, because you will have no symptoms.

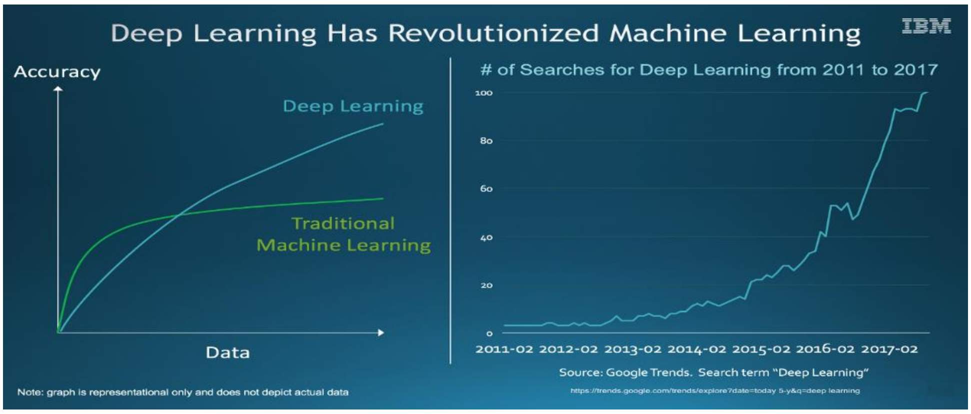
Motivation:-

The signs and the symptoms of the liver cancer are not known, till the cancer is in its advanced stage. So, early detection is the main problem. If it is detected earlier then it can be helpful for the Medical treatment to limit the danger, but it is a challenging task due to the Cancer cell structure. Interpretation of Medical image is often difficult and time consuming, even for the experienced Physicians. Most traditional medical diagnosis systems founded needs huge quantity of training data and takes long processing time. Focused on the solution to these problems, a Medical Diagnosis System based on Hidden Markov Model (HMM) is presented. This paper describes a computer aided diagnosis system for liver cancer that detects the liver tumour at an early stage from the chest CT images. This automation process reduces the time complexity and increases the diagnosis confidence. In this paper, a novel method of segmenting the CT images been discussed. This research work carried out by taking 2 CT images. The proposed work was carried out in 5 phases. In first phase, image acquisition of liver features and the second phase is related to the segmentation of ROI features of liver which can be determined using segmentation algorithm such as region growing approach. Third phase is removal of the noise. Fourth phase is feature extraction, it extract the corresponding liver nodule. Finally, the extracted liver nodules are classified. In this paper authors analyses the result for 2 images. So early detection of Liver Cancer cells can be highly possible and it reduces the risk as well. This Bio-imaging method will enhance the proper radiotherapy treatment for Liver Cancer patients. Classification is the major data mining technique which is primarily used in healthcare sectors for medical diagnosis and predicting diseases. This research work used classification algorithms namely Support Vector Machine (SVM) for liver disease prediction. Comparisons of these algorithms are done and it is based on the performance factors classification accuracy and execution time. From the experimental results, this work concludes, the SVM classifier is considered as a best algorithm because of its highest classification accuracy. On the other hand, while comparing the execution time, the Naïve Bayes classifier needs minimum execution time. E-Liang Chen et.al. [22] Computed tomography (CT) images have been widely used for liver disease diagnosis. Designing and developing computer-assisted image processing techniques to help doctors improve their diagnosis has received considerable interests over the past years. In this paper, a CT liver image diagnostic classification system is presented which will automatically find, extract the CT liver boundary and further classify liver diseases. The system comprises a detect-beforeextract (DBE) system which automatically finds the liver boundary and a neural network liver classifier which uses specially designed feature descriptors to distinguish normal liver, two types of liver tumours, hematoma and haemangioma.

Introduction of Machine Learning and Deep Learning:-

Machine Learning- Machine learning is a field of study which allows machines (computers) to learn from data or experience and make a prediction based on the experience. It enables the computers or the machines to make data-driven decisions rather than being explicitly programmed for carrying out a certain task. These programs or algorithms are designed in a way that they learn and improve over time when are exposed to new data.

Deep Learning- Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.


Deep Learning Has Revolutionized Machine Learning

Note: graph is representational only and does not depict actual data

Source: Google Trends. Search term "Deep Learning"
https://trends.google.com/trends/explore?date=today 5-y&q=deep learning

About The Dataset:

130 CT scans for segmentation of the liver as well as tumour lesions. The dataset has been split into training (80%) and test (20%) subsets provided as two separate files. The training dataset: train_data_labels_ILDS.mat contains a 463x10 matrix, where rows correspond to subjects and columns to features, and a 463x1 vector of labels. The test dataset: test_data_ILDS.mat contains a 116x10 matrix with data for the 116 test subjects. No labels are provided for the test set.

Data fields:-

1. Age: Age of the patient
2. Female: Gender of the patient (1 if Female, 0 if Male)
3. TB: Total Bilirubin
4. DB: Direct Bilirubin
5. Alkphos: Alkaline Phosphotase

6. Sgpt: Alamine Aminotransferase
7. Sgot: Aspartate Aminotransferase
8. TP: Total Protiens
9. ALB: Albumin
10. A/R: Albumin and Globulin Ratio

Dataset Link- https://www.kaggle.com/c/liver-patient-dataset/data

Project work/ Algorithm/ Design:-

Load data:-

```
liver = pd.read_csv('liver.csv')
liver.head()
```

Data Cleaning:-

```
data=liver.copy()
data=data.dropna(how='any',axis=0)
```

Model Selection:-

Train/Test Spilt:-

The widget tests learning algorithms. Different sampling schemes are available, including using separate test data. The widget does two things. First, it shows a table with different classifier performance measures, such as classification accuracy and area under the curve. Second, it outputs evaluation results, which can be used by other widgets for analysing the performance of classifiers, such as ROC Analysis or Confusion Matrix.

The widget supports various sampling methods which are as follows:

```
Train/Test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state
```

Models:-

kNN:-

The kNN gadget utilizes the kNN calculation that looks for k nearest preparing models in include space and uses their normal as expectation.

The Weights you can utilize are:

□ Uniform: all focuses in every area are weighted similarly.

□ Distance: closer neighbours of a question point have a more prominent impact than the neighbours further away

Logistic Regression:-

The logistic regression classification algorithm with LASSO (L1) or ridge (L2) regularization. Logistic Regression uses default pre-processing when no other pre-processors are given. It executes them in the following order:

- removes instances with unknown target values
- continues categorical variables (with one-hot-encoding)
- removes empty columns
- imputes missing values with mean values

Decisions Tree:-

Tree is a straightforward calculation that divides the information into hubs by class immaculateness. It is a forerunner to Random Forest. Tree in Orange is planned in-house and can deal with both discrete and ceaseless datasets. It can likewise be utilized for both grouping and relapse errands.

The parameters of tree are:

•Induce double tree
•Min. number of occurrences in leaves
•Do not split subsets less than
•Limit the maximal tree profundity

Random Forest:

An irregular timberland is an AI procedure that is utilized to tackle relapse and order issues. It uses troupe realizing, which is a strategy that joins numerous classifiers to give answers for complex issues Random Forest forms a bunch of choice trees. Each tree is created from a bootstrap test from the preparation information. When creating singular trees, a subjective subset of traits is drawn (thus the expression "Arbitrary"), from which the best characteristic for the split is chosen. The last model depends on the greater part vote from separately created trees in the woods.

The basic properties of Random Forest are:

- No of trees
- No of trees considered at each split
- Balance class distribution

AdaBoost Classifier:-

The AdaBoost (short for “Adaptive boosting”) widget is a machine-learning algorithm, formulated by Yoav Freund and Robert Schapiro. It can be used with other learning algorithms to boost their performance. It does so by tweaking the weak learners. **AdaBoost** works for both classification and regression. AdaBoost uses default pre-processing when no other pre-processors are given. It executes them in the following order:

- removes instances with unknown target values
- continuers categorical variables (with one-hot-encoding)
- removes empty columns
- imputes missing values with mean values

Gradient Boosting:-

Predict using gradient boosting on decision trees. Gradient Boosting uses default pre-processing when no other pre-processors are given. It executes them in the following order:

- removes instances with unknown target values
- continuers categorical variables (with one-hot-encoding)
- removes empty columns
- imputes missing values with mean values

Principal Component Analysis (PCA):-

The PCA linear transformation of the input data. It outputs either a transformed dataset with weights of individual instances or weights of principal components. PCA can be used to simplify visualizations of large datasets. Below, we used the *Iris* dataset to show how we can improve the visualization of the dataset with PCA. The transformed data in the Scatter Plot show a much clearer distinction between classes than the default settings.

Confusion Matrix:-

The Confusion Matrix gives the number/extent of examples between the anticipated and real class. The determination of the components in the grid takes care of the comparing occasions into the yield signal. The gadget generally gets the assessment results from Test and Score; an illustration of the blueprint is displayed underneath.

ROC Analyse:-

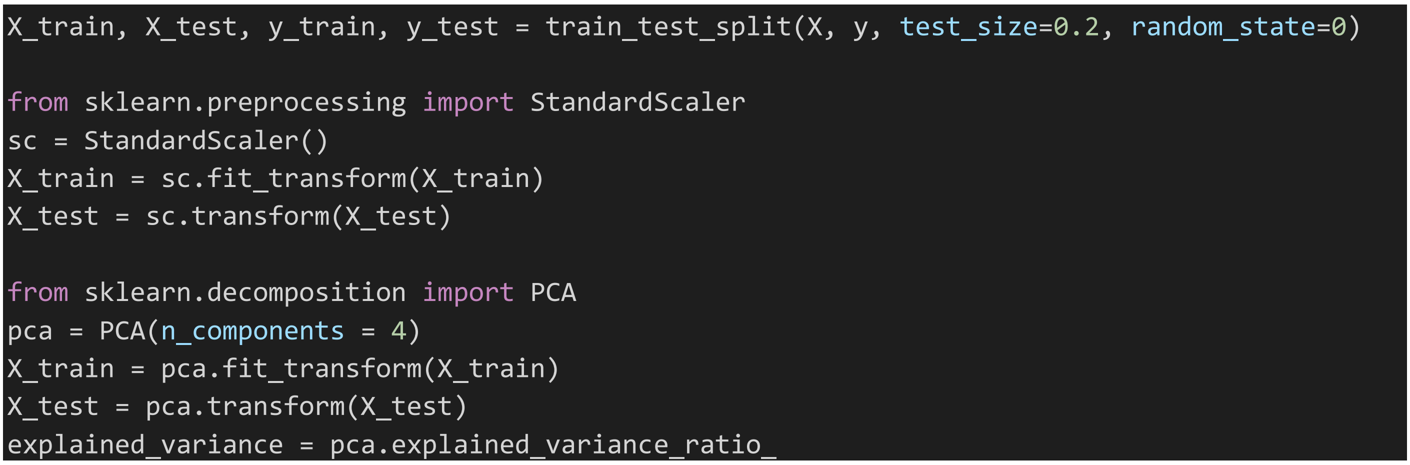
Plots a true positive rate against a false positive rate of a test.The widget shows ROC curves for the tested models and the corresponding convex hull. It serves as a mean of comparison between classification models. The curve plots a false positive rate on an x-axis (1-specificity; probability that target=1 when true value=0) against a true positive rate on a y-axis (sensitivity; probability that target=1 when true value=1). The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the classifier. Given the costs of false positives and false negatives, the widget can also determine the optimal classifier and threshold.

Bar Plot:-

Visualizes comparisons among discrete categories. The Bar Plot widget visualizes numeric variables and compares them by a categorical variable. The widget is useful for observing outliers, distributions within groups, and comparing categories. The **Bar Plot** widget is most commonly used immediately after the File widget to compare categorical values. In this example, we have used *heart-disease* data to inspect our variables.

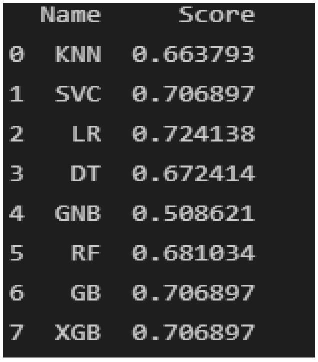
Result:-

PCA:-



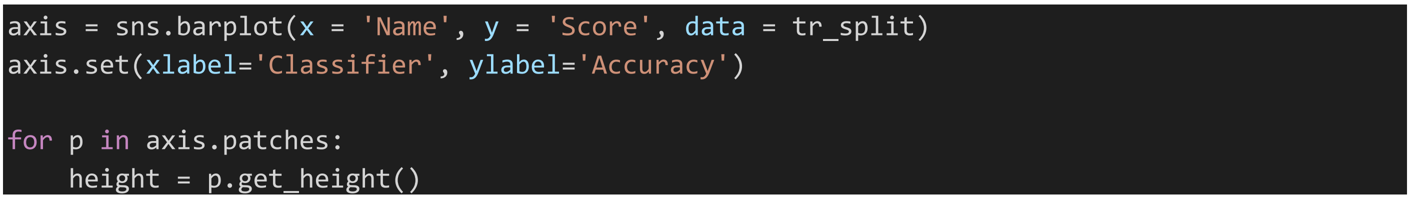
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

from sklearn.decomposition import PCA
pca = PCA(n_components = 4)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
```

Output:-



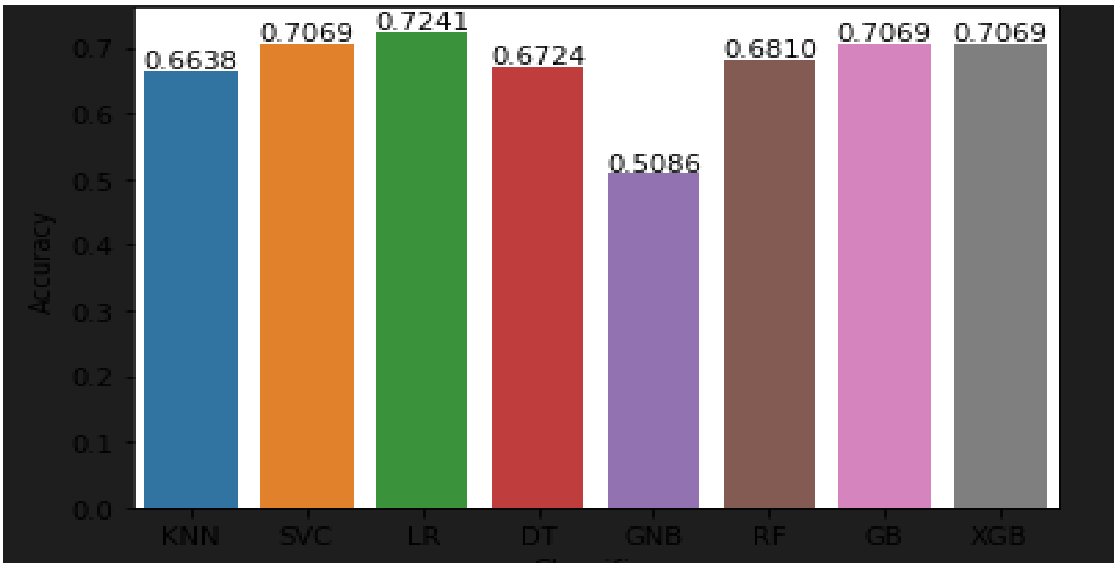
```
  Name     Score
0  KNN  0.663793
1  SVC  0.706897
2   LR  0.724138
3   DT  0.672414
4  GNB  0.508621
5   RF  0.681034
6   GB  0.706897
7  XGB  0.706897
```

In Bar Plot:-



```
axis = sns.barplot(x = 'Name', y = 'Score', data = tr_split)
axis.set(xlabel='Classifier', ylabel='Accuracy')

for p in axis.patches:
    height = p.get_height()
    axis.text(p.get_x() + p.get_width()/2, height + 0.005, '{:1.4f}'.format(height), 
ha="center")

plt.show()
```

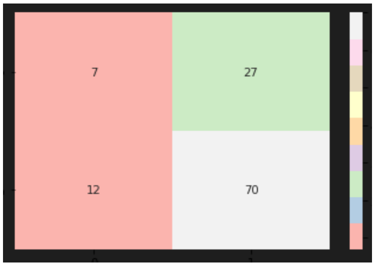
Output:-



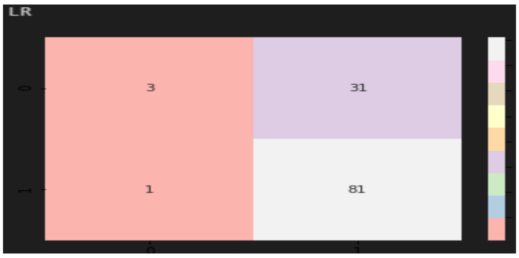
Confusion Matrix:-

```
import seaborn
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt

for name,model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    seaborn.heatmap(confusion_matrix(y_test,y_pred),annot=True,cmap='Pastel1')
    print(name)
    plt.show()
```

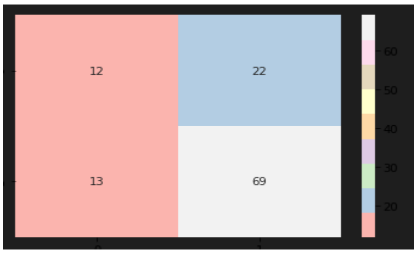
Output:-



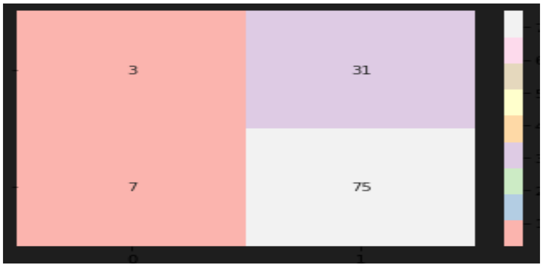
(KNN)



(LR)



(RF)

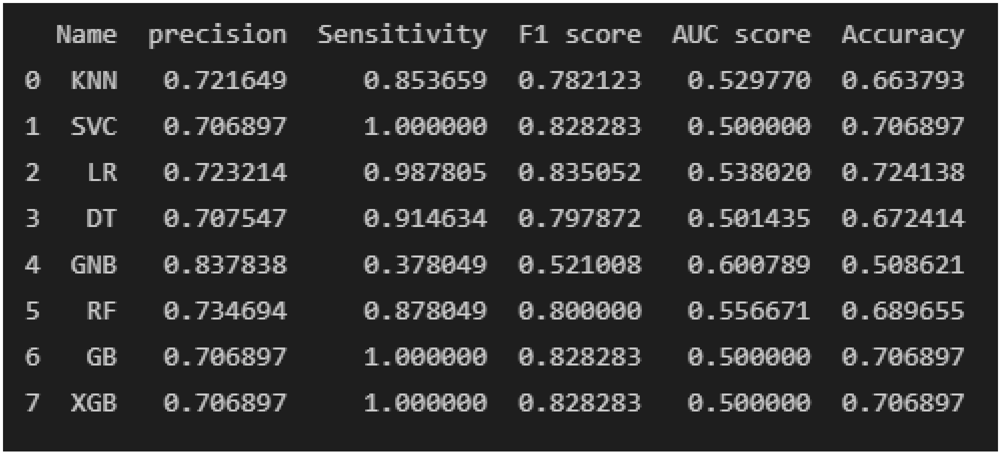


(DT)

Accuracy:-

```
from sklearn import metrics
from sklearn.metrics import f1_score,matthews_corrcoef
from sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score
fpr, tpr,_=roc_curve(model.predict(X_train),y_train,drop_intermediate=False)
names = []
AUC_score = []
precision=[]
Recall=[]
Accuracy=[]
F1_score=[]
Matthews_corrcoef=[]
specificity=[]
for name,model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    confusion = metrics.confusion_matrix(y_test, y_pred)
    TP = confusion[1, 1]
    TN = confusion[0, 0]
    FP = confusion[0, 1]
    FN = confusion[1, 0]
    Accuracy.append(accuracy_score(y_test, y_pred))
    Recall.append(metrics.recall_score(y_test, y_pred))
    precision.append(metrics.precision_score(y_test, y_pred))
    AUC_score.append(roc_auc_score(y_test, y_pred))
    F1_score.append(f1_score(y_test,y_pred))
    Matthews_corrcoef.append(matthews_corrcoef(y_test,y_pred))
    specificity.append(TN / (TN + FP))
```

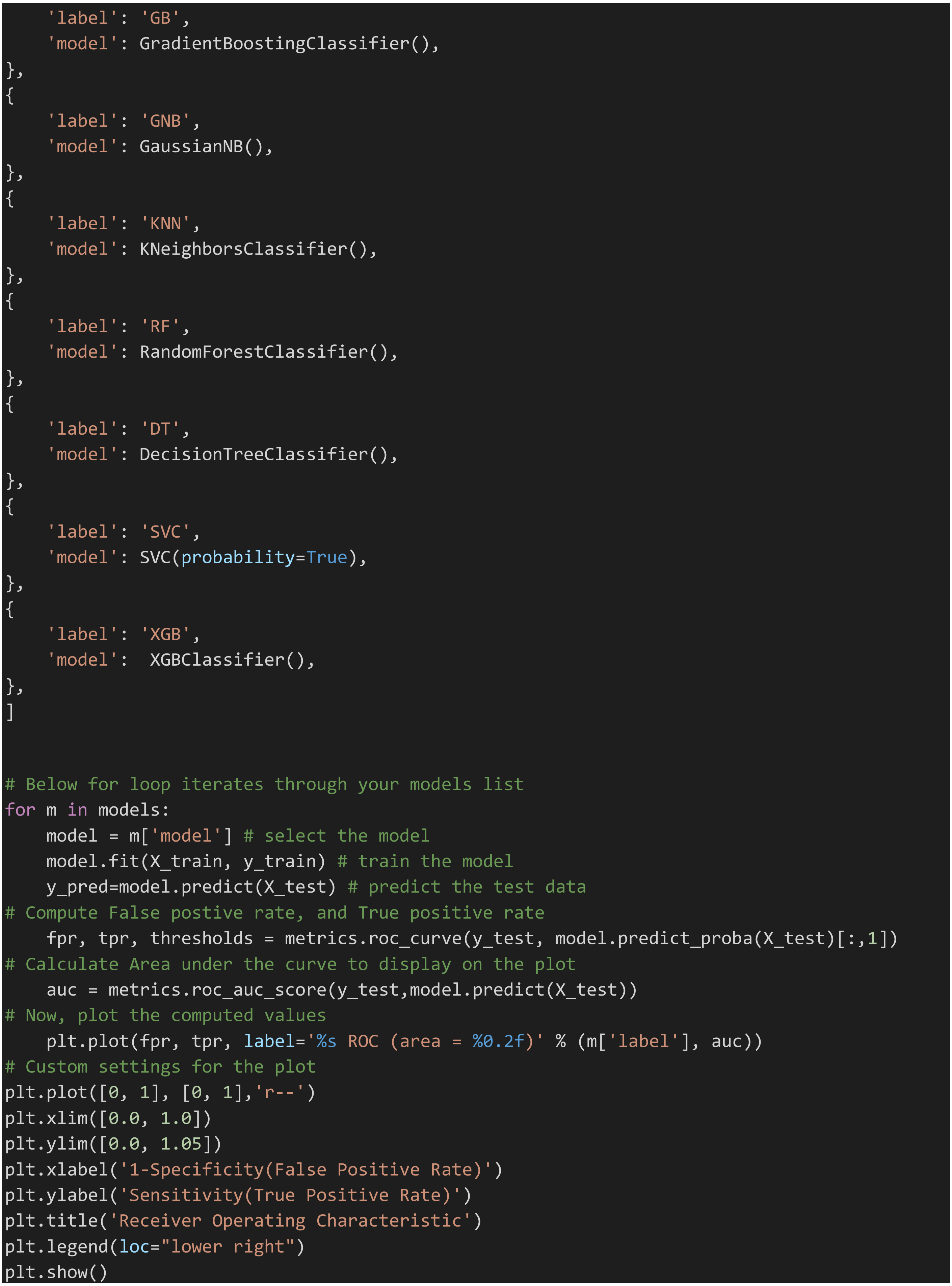
Output:-



```
  Name  precision  Sensitivity  F1 score  AUC score  Accuracy
0  KNN   0.721649     0.853659  0.782123   0.529770  0.663793
1  SVC   0.706897     1.000000  0.828283   0.500000  0.706897
2   LR   0.723214     0.987805  0.835052   0.538020  0.724138
3   DT   0.707547     0.914634  0.797872   0.501435  0.672414
4  GNB   0.837838     0.378049  0.521008   0.600789  0.508621
5   RF   0.734694     0.878049  0.800000   0.556671  0.689655
6   GB   0.706897     1.000000  0.828283   0.500000  0.706897
7  XGB   0.706897     1.000000  0.828283   0.500000  0.706897
```

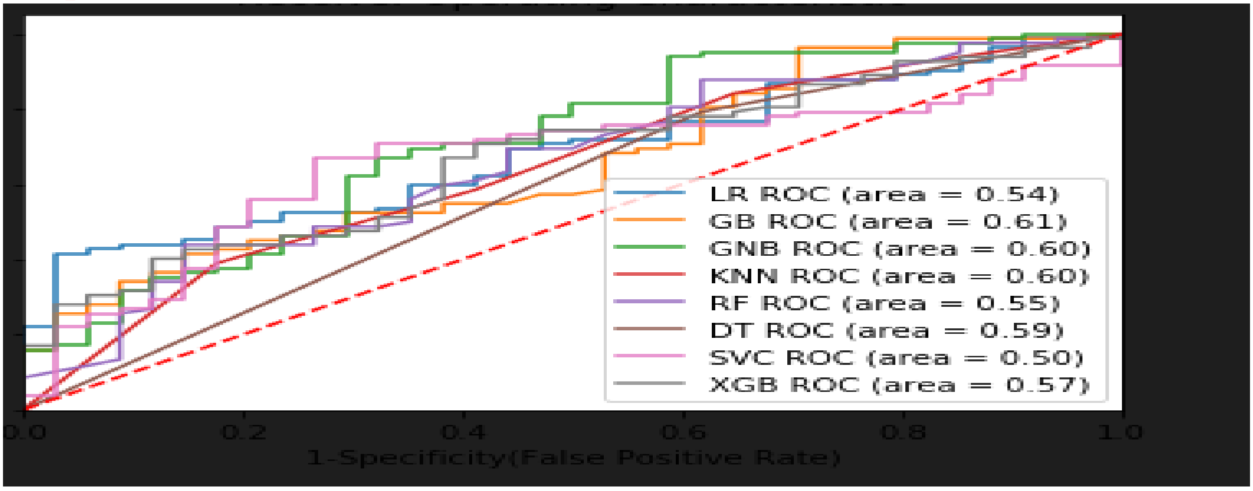
ROC Analyse:-

```
plt.figure()

# Add the models to the list that you want to view on the ROC plot
models = [
{
    'label': 'LR',
    'model': LogisticRegression(),
},
{
```



```
    'label': 'GB',
    'model': GradientBoostingClassifier(),
},
{
    'label': 'GNB',
    'model': GaussianNB(),
},
{
    'label': 'KNN',
    'model': KNeighborsClassifier(),
},
{
    'label': 'RF',
    'model': RandomForestClassifier(),
},
{
    'label': 'DT',
    'model': DecisionTreeClassifier(),
},
{
    'label': 'SVC',
    'model': SVC(probability=True),
},
{
    'label': 'XGB',
    'model':  XGBClassifier(),
},
]


# Below for loop iterates through your models list
for m in models:
    model = m['model'] # select the model
    model.fit(X_train, y_train) # train the model
    y_pred=model.predict(X_test) # predict the test data
# Compute False postive rate, and True positive rate
    fpr, tpr, thresholds = metrics.roc_curve(y_test, model.predict_proba(X_test)[:,1])
# Calculate Area under the curve to display on the plot
    auc = metrics.roc_auc_score(y_test,model.predict(X_test))
# Now, plot the computed values
    plt.plot(fpr, tpr, label='%s ROC (area = %0.2f)' % (m['label'], auc))
# Custom settings for the plot
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('1-Specificity(False Positive Rate)')
plt.ylabel('Sensitivity(True Positive Rate)')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.show()
```

Output:-



Discussion:-

This paper gives survey on different types of Machine Learning Techniques used in liver cancer analysis. However the liver tumor is difficult to detect from the CT or MRI images because of two reasons: one is the difference in the liver and non-liver pixel intensities in CT images and another one is detection of liver from overlapped organs. Hence segmentation helps doctors to provide effective treatment by knowing nature of the tumour. The first paper focused on the solution to these problems, a Medical Diagnosis System based on Hidden Markov Model (HMM) is presented. This paper describes a computer aided diagnosis system for liver cancer that detects the liver tumour at an early stage from the chest CT images. Second paper employ a support vector machine (SVM) classifier, which is trained using the user fed image, sets, to classify the tumour region from liver image. Third paper algorithms used in this work are Naïve Bayes and support vector machine (SVM). These classifier algorithms are compared based on the performance factors i.e. classification accuracy and execution time. From the experimental results it is observed that the SVM is a better classifier for predict the liver diseases. Fourth paper describes a modified probabilistic neural network (PNN) [MPNN] in conjunction with feature descriptors which are generated by fractal feature information and the grey-level co-occurrence matrix. ML algorithms can automatically extract imaging features and identify optimal subsets of features from large data sets, particularly when combined with radionics' analysis. Relative to traditional statistical models, ML models demonstrate improved predictive performance in the prognostic study of HCC. Regrettably, most existing ML predictive models lack external validation, which is an obstacle to serving HCC patients as personalized predictive tools. Although most current ML algorithms are preliminary, this promising method will be widely accepted in clinical practice in the future.

Reference:-

[1]. Liver- human body organs, [online], available: www.organsofthebody.com accessed: [12-01-2016].

[2]. Liver, liver function, pain location, cause and symptoms [online], available: www.ihealthblogger.com accessed: [12-01-2016].

[3]. Anatomy and functions of liver, [online], available: www.stanford.com accessed: [14-01-2016].

[4]. Liver definition structure [online], available: www.wikipedia.com accessed: [18-01-2016].

[5]. Body maps and organs [online], available: www.healthline.com accessed: [19-01-2016].

[6]. Human liver workhorse of the body, [online], available: www.britannica.com accessed: [22-01-2016].

Dataset Link- https://www.kaggle.com/c/liver-patient-dataset/data