**LIVER DISEASE PREDICTION**

1. **ABSTRACT:**

In recent years in Healthcare sectors, machine learning became an ease of use for disease prediction. It's awfully difficult task to the researchers to predict the diseases from the voluminous medical databases. To beat this issue the researchers use machine learning techniques like classification, clustering, association rules. Identification of disease at a preliminary stage is very important for higher treatment. It's a difficult task for medical researchers to predict the illness within the early stages because of delicate symptoms. It is a very challenging task for medical researchers to predict the disease in the early stages owing to subtle symptoms. Often the symptoms become apparent when it is too late. To overcome this issue, this project aims to improve disease diagnosis using machine learning approaches. The objective of this project is to use classification algorithms to spot the liver disease patients from healthy people. This project also aims to compare the classification algorithms based on their performance factors i.e. classification accuracy and execution time. We all know that there has been a rise in disease because of varied factors. By using the liver disease dataset we are going to come up with the best prediction algorithm to help doctors with their work. In order to do this we are going to have to look into the data and use different classification methods to find what works best for the data we have. Classification techniques are much popular in medical diagnosis and predicting diseases. One of the classification technique is SVM. SVM aims to find an optimal hyper plane that separates the data into different classes. The scikit-learn package in python is used for implementing SVM. The pre-processed data is split into test data and training set which is of 25% and 75% of the total dataset respectively. A support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space. A good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

1. **INTRODUCTION:**

The healthcare sector has long been an early adopter of and benefited greatly from technological advances. These days, machine learning (a subset of [artificial intelligence](https://builtin.com/artificial-intelligence)) plays a key role in many health-related realms, including the development of new medical procedures, the handling of patient data and records and the treatment of chronic diseases.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiringunlabeled data generally doesn’t require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

The increasingly growing number of applications of machine learning in healthcare allows us to glimpse at a future where data, analysis, and innovation work hand-in-hand to help countless patients without them ever realizing it. Soon, it will be quite common to find ML-based applications embedded with real-time patient data available from different healthcare systems in multiple countries, thereby increasing the efficacy of new treatment options which were unavailable before.

One of the chief ML applications in healthcare is the identification and diagnosis of diseases and ailments which are otherwise considered hard-to-diagnose. This can include anything from cancers which are tough to catch during the initial stages, to other genetic diseases. IBM Watson Genomics is a prime example of how integrating cognitive computing with genome-based tumor sequencing can help in making a fast diagnosis. Berg, the biopharma giant is leveraging AI to develop therapeutic treatments in areas such as oncology. P1vital's PReDicT (Predicting Response to Depression Treatment) aims to develop a commercially feasible way to diagnose and provide treatment in routine clinical conditions.

One of the primary clinical applications of machine learning lies in early-stage drug discovery process. This also includes R&D technologies such as next-generation sequencing and precision medicine which can help in finding alternative paths for therapy of multifactorial diseases. Currently, the machine learning techniques involve unsupervised learning which can identify patterns in data without providing any predictions. Project Hanover developed by Microsoft is using ML-based technologies for multiple initiatives including developing AI-based technology for cancer treatment and personalizing drug combination for AML (Acute Myeloid Leukemia).

Machine learning and deep learning are both responsible for the breakthrough technology called Computer Vision. This has found acceptance in the InnerEye initiative developed by Microsoft which works on image diagnostic tools for image analysis. As machine learning becomes more accessible and as they grow in their explanatory capacity, expect to see more data sources from varied medical imagery become a part of this AI-driven diagnostic process.

Personalized treatments can not only be more effective by pairing individual health with predictive analytics but is also ripe are for further research and better disease assessment. Currently, physicians are limited to choosing from a specific set of diagnoses or estimate the risk to the patient based on his symptomatic history and available genetic information. But machine learning in medicine is making great strides, and IBM Watson Oncology is at the forefront of this movement by leveraging patient medical history to help generate multiple treatment options. In the coming years, we will see more devices and biosensors with sophisticated health measurement capabilities hit the market, allowing more data to become readily available for such cutting-edge ML-based healthcare technologies.

Behavioural modification is an important part of preventive medicine, and ever since the proliferation of machine learning in healthcare, countless startups are cropping up in the fields of cancer prevention and identification, patient treatment, etc. Somatix is a B2B2C-based data analytics company which has released an ML-based app to recognize gestures which we make in our daily lives, allowing us to understand our unconscious behaviour and make necessary changes.

Maintaining up-to-date health records is an exhaustive process, and while technology has played its part in easing the data entry process, the truth is that even now, a majority of the processes take a lot of time to complete. The main role of machine learning in healthcare is to ease processes to save time, effort, and money. Document classification methods using vector machines and ML-based OCR recognition techniques are slowly gathering steam, such as Google's Cloud Vision API and MATLAB's machine learning-based handwriting recognition technology. MIT is today at the cutting edge of developing the next generation of intelligent, smart health records, which will incorporate ML-based tolls from the ground up to help with diagnosis, clinical treatment suggestions, etc.

Machine learning has several potential applications in the field of clinical trials and research. As anybody in the pharma industry would tell you, clinical trials cost a lot of time and money and can take years to complete in many cases. Applying ML-based predictive analytics to identify potential clinical trial candidates can help researchers draw a pool from a wide variety of data points, such as previous doctor visits, social media, etc. Machine learning has also found usage in ensuring real-time monitoring and data access of the trial participants, finding the best sample size to be tested, and leveraging the power of electronic records to reduce data-based errors.

Crowdsourcing is all the rage in the medical field nowadays, allowing researchers and practitioners to access a vast amount of information uploaded by people based on their own consent. This live health data has great ramifications in the way medicine will be perceived down the line. Apple's Research Kit allows users to access interactive apps which apply ML-based facial recognition to try and treat Asperger's and Parkinson's disease. IBM recently partnered with Medtronic to decipher, accumulate, and make available diabetes and insulin data in real time based on the crowd sourced information. With the advancements being made in IoT, the healthcare industry is still discovering new ways in which to use this data and tackle tough-to-diagnose cases and help in the overall improvement of diagnosis and medication.

One of the most sought-after applications of machine learning in healthcare is in the field of Radiology. Medical image analysis has many discrete variables which can arise at any particular moment of time. There are many lesions, cancer foci, etc. which cannot be simply modelled using complex equations. Since ML-based algorithms learn from the multitude of different samples available on-hand, it becomes easier to diagnose and find the variables. One of the most popular uses of machine learning in medical image analysis is the classification of objects such as lesions into categories such as normal or abnormal, lesion or non-lesion, etc. Google's Deep Mind Health is actively helping researchers in UCLH develop algorithms which can detect the difference between healthy and cancerous tissue and improve radiation treatment for the same.

AI-based technologies and machine learning are today also being put to use in monitoring and predicting epidemics around the world. Today, scientists have access to a large amount of data collected from satellites, real-time social media updates, website information, etc. Artificial neural networks help to collate this information and predict everything from malaria outbreaks to severe chronic infectious diseases. Predicting these outbreaks is especially helpful in third-world countries as they lack in crucial medical infrastructure and educational systems. A primary example of this is the ProMED-mail, an Internet-based reporting platform which monitors evolving diseases and emerging ones and provides outbreak reports in real-time.

The liver is an immense, significant organ in the human body weights around 3 pounds. The liver contains two huge portions, called the privilege and the left projections. The gallbladder sits under the liver, nearby parts of the pancreas and stomach related organs. The liver and these organs cooperate to process, ingest, and process sustenance. The liver fundamental job is to channel the hurtful substances in the blood starting from the stomach related framework, before passing it to whatever is left of the body. Liver" disease can be acquired either by outside factors like infections and liquor use or through genetics. Corpulence has additionally been related with this illness. Overtime harm to liver can cause liver failure or at some circumstances even life-threatening condition. Liver disease is one of the most "death-dealing disease on the planet. The fundamental driver of liver harm are Fatty liver, Liver Fibrosis, Cirrhosis, hepatitis and diseases.

In the beginning times of liver illness, it is exceptionally hard to identify even though liver tissue has already been harmed. It requires numerous specialists to analyse the damage.

This can contort pharmaceutical medications, so early finding is essential to spare the patient. Common Liver Disorders include:

* Fatty liver is an agonizing liver condition portrayed by liver irritation and arrangement of scar tissue, which has numerous conceivable causes, including corpulence, poor nourishment and certain meds, among numerous others. It can happen in individuals with an abnormal state of liquor utilization as well as in individuals who never had liquor.
* Cirrhosis is another important type of liver damage. It is usually the result of long term damage of liver. When liver is damaged for a long time and starts to malfunction this particular type of liver damage occurs.
* Hepatitis is usually caused by an infection that spreads by direct contact with tainted body
* Liver Cancer risk is higher on those who has cirrhosis. Most often it spreads from liver to other organs.
  1. **PROBLEM STATEMENT:**

Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. A sample dataset used to evaluate prediction algorithms in an effort to reduce burden on doctors. The dataset contains various attributes of 583 Indian patients, define a classification algorithm which can identify whether a person is suffering from liver disease or not.

* 1. **OBJECTIVE:**

The objective of the project is to predict whether a patient is having a liver disease or not from a sample dataset, and should be able to calculate the predictability to the greater magnitude of accuracy using the appropriate machine learning algorithm.

* 1. **PROPOSED SYSTEM:**

Machine learning is one amongst the foremost extensively utilised paradigms of huge data management wherever a considerably high set of distinct data may be collated effectively to form applicable inferences and eventually to come back up with a usual assortment of contextually helpful collection of integrative data. With the onset of the exponential technological explosion within the field of drugs, there's a felt have to be compelled to handle an enormous set of information, thereby managing and utilizing constant to form effective and informative inferences for the doctors and patients.

* 1. **ADVANTAGES IN PROPOSED SYSTEM:**
* The performance classification of liver-based diseases is further improved.
* By using Machine learning models time complexity and accuracy can be measured, so by the needs of the user we can measure the various parameters.
* The various machine learning algorithms used can have high accuracy of the result.
* Risk factors can be predicted early by machine learning algorithms

1. **BLOCK DIAGRAM:**

Data cleaning and visualization

Read the dataset

Dividing the data in to train, test and validate sets

Feature selection

Random forest classifier

SVM classifier

KNN classifier

K-fold cross validation selection

Performance analysis of the classifier

Classifier selection

Performance and accuracy results

1. **MODULES:**
2. **Exploratory data analysis:**

Through this process we can figure out general looking at the data to know whats going on. Inspect the data in the dataset: that means it checks whether there is any missing data or any irrelevant data and do a cleanup for the data.

Databases of 583 records/entries are taken from the ILPD (Indian Liver Patient Dataset). The dataset is downloaded from UCI Machine Learning Repository. Entire ILPD dataset contains info regarding 583 Indian liver patients. within which 416 are liver patient records and 167 non liver patient records .The data set was collected from north east of province, India. Selector could be a category label wont to divide into teams (liver patient or not)

### Describing the data:

### By using the method describe(), we can calculate parameters like count, mean, std, and max.

### Shape of the dataset:

It gives the dimension of the dataset, i.e; number of columns and rows the dataset contains.

### Extracting data from the dataset:

### To display first 10 rows of the data from the dataset, we can call the head() method on it. To this, we can pass it the argument 10.

### Performing operations around a variable:

We will perform some operations on a variable. for example, here, we demonstrate how to group a data on a variable. For this, we tend to use the groupby() perform.

1. **Data visualization**

From the selected dataset, we have 583 rows of values, 10 attributes and an independent variable. We implement Imputer to find the missing data, encoding the categorical data, Feature Scaling and Dimesnionality Reduction using Principle Component Analysis (PCA) method.

In this phase we will visualize the number of patients diagnosed with liver disease and patents not diagnosed with liver disease. We can identify the number of patients that are male,female and we will visualize their age .

We can use pandas with Matplotlib for visualizing data.

### Histograms

### Since histograms group information into bins and give us a thought of what number of perceptions each bin holds, this is a good method to visualize information for ML. The shapes of the bins disclose to us whether a quality is Gaussian, skewed, or has an exponential dispersion. It likewise implies us about outliers.

### Density Plots

### A density plot appears to be an abstracted histogram. Each bin has a smooth bend drawn through its top.

1. **Box and Whisker Plots**

A box plot summarizes how each attribute is dispersed. It likewise draws a line for the median and a box around the 25th and 75th percentiles. whiskers reveal to us how the information is spread, and the dots outside the whiskers give candidate outlier values.

**Visualization for multivariate plots**

1. **Correlation Matrix Plot**

Such a plot signifies how changes between two variables relate. Two variables that adjustment a similar way are emphatically related. A change in inverse ways suggests negative correlation.

1. **Scatterplot Matrix**

Scatterplot networks delineate how two variables relate as spots in two measurements. Plotting all scatterplots for an information together in one spot brings about a scatterplot grid. These plots can spot organized connections between variables.

## Divide the Dataset into Training set and Test Set

Presently we have to split our dataset into two sets— a Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general principle of the thumb is to distribute 80% of the dataset to training set and the staying 20% to test set. For this undertaking, we will import test\_train\_split from model\_selection library of scikit.

Now to build our training and test sets, we will create 4 sets— X\_train (training part of the matrix of features), X\_test (test part of the matrix of features), Y\_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices) , Y\_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices). We will allot to them the test\_train\_split, which takes the parameters — exhibits (X and Y), test\_size (on the off chance that we give it the worth 0.5, which means half, it would part the dataset into half. Since a perfect decision is to apportion 20% of the dataset to test set, it is normally assigned as 0.2. 0.25 would mean 25%, simply saying).

1. **Implementing classifiers:**

The classification algorithms we used in our project are:

1. K-Nearest Neighbors
2. Support Vector Machines
3. Random Forest

**K-Nearest Neighbors**

The k-nearest-neighbors algorithm is a classification algorithm, and it is supervised: it takes a lot of marked points and uses them figure out how to name different other points. To label another point, it looks at the labelled points closest to that new point (those are its nearest neighbors), and has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point (the “k” is the number of neighbors it checks).

## Support Vector Machine

SVM or Support Vector Machine is a linear model for classification and regression problems. It can tackle linear and non-linear problems and work well for many reasonable issuess. The idea of SVM is simple: The algorithm creates a line or a hyperplane which sisolates the data into classes.

As per SVM algorithm we discover the points nearest to the line from both the classes.These points are called support vectors. Now, we figure the distance between the line and the support vectors. This distance is called the margin. Our main goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane

Thus SVM tries to make a decision boundary in such a way that the separation between the two classes is as wide as possible.

## ****Random Forest:****

Random forest is a type of supervised machine learning algorithm based on [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning). Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The [random forest](https://en.wikipedia.org/wiki/Random_forest) algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees*,* hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees’ habit of over fitting to their training set.

1. **K-fold cross validation:**

To ensure that our model isn't over fitting when it is executed to the new arrangement of test information, we at that point actualized K-Fold Cross Validation to know the mean exactness and the standard deviation from given 10 diverse arrangement of approval.

Separation the Dataset into Training set and Test Set: Now we have to part our dataset into two sets — a Training set and a Test set. We will prepare our AI models on our preparation set, i.e our AI models will attempt to see any relationships in our preparation set and afterward we will test the models on our test set to check how precisely it can foresee. A general standard of the thumb is to dispense 80% of the dataset to preparing set and the staying 20% to test set. For this task, we will import test\_train\_split from model\_selection library of scikit.

Presently to assemble our preparation and test sets, we will make 4 sets—X\_train (preparing some portion of the framework of highlights), X\_test (test some portion of the network of highlights), Y\_train (preparing some portion of the needy factors related with the X train sets, and along these lines additionally a similar files) , Y\_test (test some portion of the reliant factors related with the X test sets, and in this way likewise a similar records). We will appoint to them the test\_train\_split, which takes the parameters — clusters (X and Y), test\_size.

1. **IMPLEMENTATION:**

# coding: utf-8

# In[305]:

# for numerical computing

import numpy as np

# for dataframes

import pandas as pd

# for easier visualization

import seaborn as sns

# for visualization and to display plots

from matplotlib import pyplot as plt

get\_ipython().run\_line\_magic('matplotlib', 'inline')

# import color maps

from matplotlib.colors import ListedColormap

# Ignore Warnings

import warnings

warnings.filterwarnings("ignore")

from math import sqrt

# to split train and test set

from sklearn.model\_selection import train\_test\_split

# to perform hyperparameter tuning

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.cross\_validation import cross\_val\_score

# Machine Learning Models

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.metrics import roc\_curve, auc, roc\_auc\_score, confusion\_matrix

from sklearn.preprocessing import StandardScaler

from sklearn.cross\_validation import train\_test\_split

from sklearn.cross\_validation import cross\_val\_score

from sklearn.neighbors import KNeighborsClassifier

from matplotlib.colors import ListedColormap

from sklearn.metrics import accuracy\_score

#import xgboost

import os

# to save the final model on disk

from sklearn.externals import joblib

# In[306]:

df=pd.read\_csv("E:\TARP\liver.csv")

# In[307]:

df.shape

# In[308]:

df.columns

# In[309]:

df.head()

# In[310]:

df.dtypes[df.dtypes=='object']

# In[311]:

# Plot histogram grid

df.hist(figsize=(15,15), xrot=-45, bins=10) ## Display the labels rotated by 45 degress

# Clear the text "residue"

plt.show()

# In[312]:

df.describe()

# In[313]:

## if score==negative, mark 0 ;else 1

def partition(x):

if x == 2:

return 0

return 1

df['Dataset'] = df['Dataset'].map(partition)

# In[314]:

df.describe(include=['object'])

# In[315]:

plt.figure(figsize=(5,5))

sns.countplot(y='Gender', data=df)

# In[316]:

df[df['Gender'] == 'Male'][['Dataset', 'Gender']].head()

# In[317]:

sns.factorplot (x="Age", y="Gender", hue="Dataset", data=df);

# In[318]:

sns.countplot(data=df, x = 'Gender', label='Count')

M, F = df['Gender'].value\_counts()

print('Number of patients that are male: ',M)

print('Number of patients that are female: ',F)

# In[319]:

## if score==negative, mark 0 ;else 1

def partition(x):

if x =='Male':

return 0

return 1

df['Gender'] = df['Gender'].map(partition)

# In[320]:

sns.set\_style('whitegrid') ## Background Grid

sns.FacetGrid(df, hue = 'Dataset', size = 5).map(plt.scatter, 'Total\_Bilirubin', 'Direct\_Bilirubin').add\_legend()

# In[321]:

sns.set\_style('whitegrid') ## Background Grid

sns.FacetGrid(df, hue = 'Dataset', size = 5).map(plt.scatter, 'Total\_Bilirubin', 'Albumin').add\_legend()

# In[322]:

sns.set\_style('whitegrid') ## Background Grid

sns.FacetGrid(df, hue = 'Dataset', size = 5).map(plt.scatter, 'Total\_Protiens', 'Albumin\_and\_Globulin\_Ratio').add\_legend()

# In[323]:

df.corr()

# In[324]:

plt.figure(figsize=(10,10))

sns.heatmap(df.corr())

# In[325]:

mask=np.zeros\_like(df.corr())

mask[np.triu\_indices\_from(mask)] = True

plt.figure(figsize=(10,10))

with sns.axes\_style("white"):

ax = sns.heatmap(df.corr()\*100, mask=mask, fmt='.0f', annot=True, lw=1, cmap=ListedColormap(['red', 'blue', 'green','yellow']))

# In[326]:

df = df.drop\_duplicates()

# In[327]:

sns.boxplot(df.Aspartate\_Aminotransferase)

# In[328]:

df.Aspartate\_Aminotransferase.sort\_values(ascending=False).head()

# In[329]:

df = df[df.Aspartate\_Aminotransferase <=3000 ]

df.shape

# In[330]:

sns.boxplot(df.Aspartate\_Aminotransferase)

# In[331]:

df.Aspartate\_Aminotransferase.sort\_values(ascending=False).head()

# In[332]:

df = df[df.Aspartate\_Aminotransferase <=2500 ]

df.shape

# In[333]:

df.isnull().values.any()

# In[334]:

df=df.dropna(how='any')

# In[335]:

df.shape

# In[336]:

df.head()

# In[337]:

# Create separate object for target variable

y = df.Dataset

# Create separate object for input features

X = df.drop('Dataset', axis=1)

# In[338]:

# Split X and y into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.2,

random\_state=1234,

stratify=df.Dataset)

# In[339]:

# Print number of observations in X\_train, X\_test, y\_train, and y\_test

print(X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape)

# In[340]:

train\_mean = X\_train.mean()

train\_std = X\_train.std()

# In[341]:

## Standardize the train data set

X\_train = (X\_train - train\_mean) / train\_std

# In[342]:

## Check for mean and std dev.

X\_train.describe()

# In[343]:

## Note: We use train\_mean and train\_std\_dev to standardize test data set

X\_test = (X\_test - train\_mean) / train\_std

# In[344]:

## Check for mean and std dev. - not exactly 0 and 1

X\_test.describe()

# In[345]:

tuned\_params = {'n\_estimators': [100, 200, 300, 400, 500], 'min\_samples\_split': [2, 5, 10], 'min\_samples\_leaf': [1, 2, 4]}

classifier = RandomizedSearchCV(RandomForestClassifier(), tuned\_params, n\_iter=15, scoring = 'roc\_auc', n\_jobs=-1)

classifier.fit(X\_train, y\_train)

# In[346]:

classifier.best\_estimator\_

# In[347]:

y\_train\_pred = model.predict(X\_train)

# In[348]:

y\_pred = model.predict(X\_test)

# In[349]:

# Get just the prediction for the positive class (1)

# Display first 10 predictions

y\_pred\_proba[:10]

# In[350]:

confusion\_matrix(y\_test, y\_pred).T

# In[351]:

acc\_r = accuracy\_score(y\_test, y\_pred, normalize=True) \* float(100) ## get the accuracy on testing data

acc\_r

# In[352]:

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_pred))

# In[353]:

# Calculate ROC curve from y\_test and pred

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_proba)

# In[354]:

# Plot the ROC curve

fig = plt.figure(figsize=(8,8))

plt.title('Receiver Operating Characteristic')

# Plot ROC curve

plt.plot(fpr, tpr, label='l1')

plt.legend(loc='lower right')

# Diagonal 45 degree line

plt.plot([0,1],[0,1],'k--')

# Axes limits and labels

plt.xlim([-0.1,1.1])

plt.ylim([-0.1,1.1])

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

# In[355]:

# Calculate AUC for Train set

roc\_auc\_score(y\_train, y\_train\_pred)

# In[356]:

# Calculate AUC for Test set

print(auc(fpr, tpr))

# In[357]:

## Building the model again with the best hyperparameters

classifier = RandomForestClassifier(n\_estimators=500, min\_samples\_split=2, min\_samples\_leaf=4)

classifier.fit(X\_train, y\_train)

# In[358]:

# creating odd list of K for KNN

neighbors = list(range(1,20,2))

# empty list that will hold cv scores

cv\_scores = []

# 10-fold cross validation , 9 datapoints will be considered for training and 1 for cross validation (turn by turn) to determine value of k

for k in neighbors:

knn = KNeighborsClassifier(n\_neighbors=k)

scores = cross\_val\_score(knn, X\_train, y\_train, cv=5, scoring='accuracy')

cv\_scores.append(scores.mean())

# changing to misclassification error

MSE = [1 - x for x in cv\_scores]

# determining best k

optimal\_k = neighbors[MSE.index(min(MSE))]

print('\nThe optimal number of neighbors is %d.' % optimal\_k)

# In[359]:

MSE.index(min(MSE))

# In[360]:

# plot misclassification error vs k

plt.plot(neighbors, MSE)

plt.xlabel('Number of Neighbors K')

plt.ylabel('Misclassification Error')

plt.show()

# In[361]:

classifier = KNeighborsClassifier(n\_neighbors = optimal\_k)

classifier.fit(X\_train, y\_train)

# In[362]:

y\_pred = classifier.predict(X\_test)

y\_train\_pred = classifier.predict(X\_train)

# In[363]:

acc\_k = accuracy\_score(y\_test, y\_pred, normalize=True) \* float(100) ## get the accuracy on testing data

acc\_k

# In[364]:

cnf=confusion\_matrix(y\_test,y\_pred).T

cnf

# In[365]:

print(classification\_report(y\_test,y\_pred))

# In[366]:

# Get just the prediction for the positive class (1)

y\_pred\_proba = classifier.predict\_proba(X\_test)[:,1]

# In[367]:

# Display first 10 predictions

y\_pred\_proba[:10]

# In[368]:

# Calculate ROC curve from y\_test and pred

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_proba)

# In[369]:

# Plot the ROC curve

fig = plt.figure(figsize=(8,8))

plt.title('Receiver Operating Characteristic')

# Plot ROC curve

plt.plot(fpr, tpr, label='l1')

plt.legend(loc='lower right')

# Diagonal 45 degree line

plt.plot([0,1],[0,1],'k--')

# Axes limits and labels

plt.xlim([-0.1,1.1])

plt.ylim([-0.1,1.1])

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

# In[370]:

# Calculate AUC for Train

roc\_auc\_score(y\_train, y\_train\_pred)

# In[371]:

# Calculate AUC for Test

print(auc(fpr, tpr))

# In[372]:

from sklearn import svm

def svc\_param\_selection(X, y, nfolds):

Cs = [0.001, 0.01, 0.1, 1, 10]

gammas = [0.001, 0.01, 0.1, 1]

param\_grid = {'C': Cs, 'gamma' : gammas}

grid\_search = GridSearchCV(svm.SVC(kernel='rbf'), param\_grid, cv=nfolds)

grid\_search.fit(X\_train, y\_train)

grid\_search.best\_params\_

return grid\_search.best\_params\_

# In[373]:

classifier=SVC(kernel='rbf',probability=True)

classifier.fit(X\_train,y\_train)

# In[374]:

svc\_param\_selection(X\_train,y\_train,5)

# In[375]:

###### Building the model again with the best hyperparameters

classifier= SVC(C=1, gamma=1)

classifier.fit(X\_train, y\_train)

# In[376]:

## Predict Train results

y\_train\_pred = model.predict(X\_train)

## Predict Test results

y\_pred = model.predict(X\_test)

# In[377]:

confusion\_matrix(y\_test, y\_pred).T

# In[378]:

acc\_s = accuracy\_score(y\_test, y\_pred, normalize=True) \* float(100) ## get the accuracy on testing data

acc\_s

# In[379]:

print(classification\_report(y\_test,y\_pred))

# In[380]:

# Calculate ROC curve from y\_test and pred

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_proba)

# In[381]:

# Plot the ROC curve

fig = plt.figure(figsize=(8,8))

plt.title('Receiver Operating Characteristic')

# Plot ROC curve

plt.plot(fpr, tpr, label='l1')

plt.legend(loc='lower right')

# Diagonal 45 degree line

plt.plot([0,1],[0,1],'k--')

# Axes limits and labels

plt.xlim([-0.1,1.1])

plt.ylim([-0.1,1.1])

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

# In[382]:

# Calculate AUC for Train

roc\_auc\_score(y\_train, y\_train\_pred)

# In[383]:

print(auc(fpr, tpr))

# In[384]:

models\_comparison = [

['KNN',acc\_k],

['Support Vector Classfication',acc\_s],

['Random Forest Classifiaction',acc\_r]

]

models\_compaison\_df = pd.DataFrame(models\_comparison,columns=['Model','% Accuracy'])

models\_compaison\_df.head()

# In[385]:

from sklearn.model\_selection import cross\_val\_score

accuracies = cross\_val\_score(estimator = classifier,

X = X\_train, y = y\_train,

cv = 10, n\_jobs = -1)

print("Showing all 10 of K-Fold Cross Validation accuracies:\n", accuracies)

accuracies\_mean = accuracies.mean()

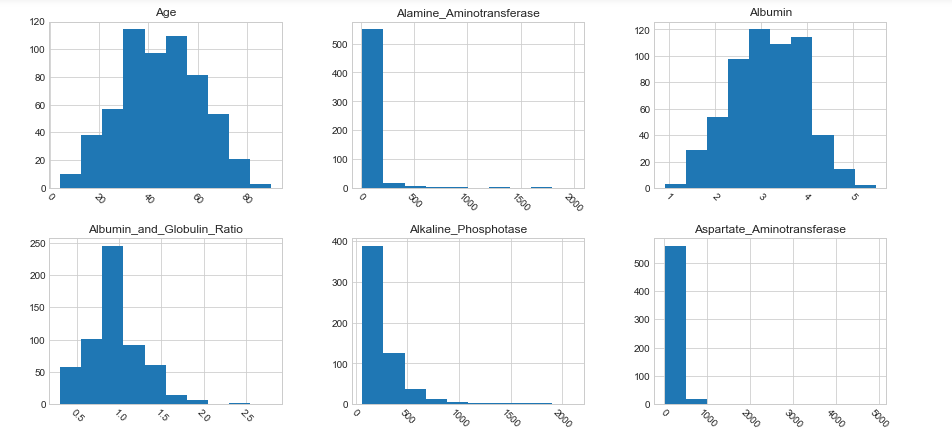
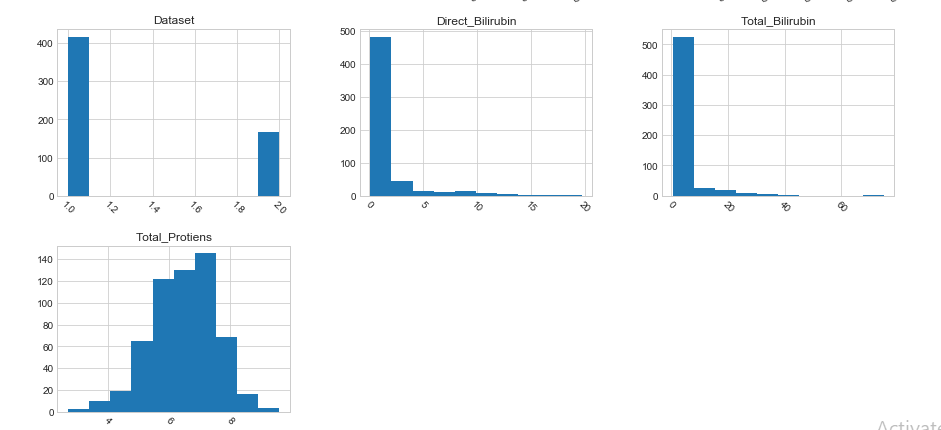
print("\nMean of accuracies:\n", accuracies\_mean)

accuracies\_std = accuracies.std()

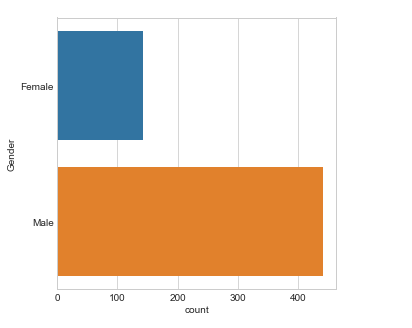
print("\nStandard Deviation:\n", accuracies\_std)

1. **RESULTS:**

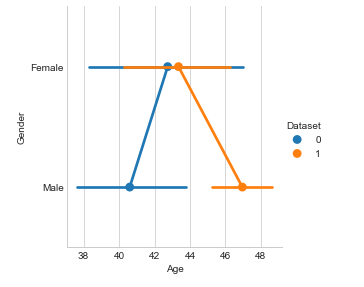
**Distribution of numerical features:**

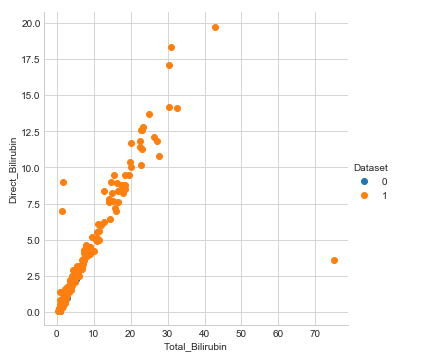
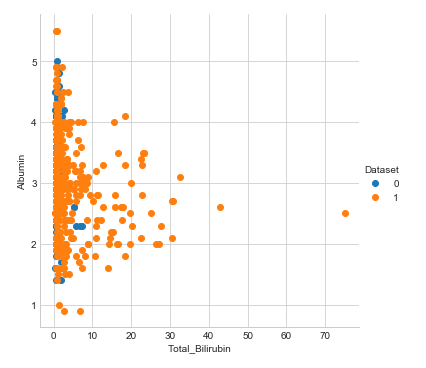
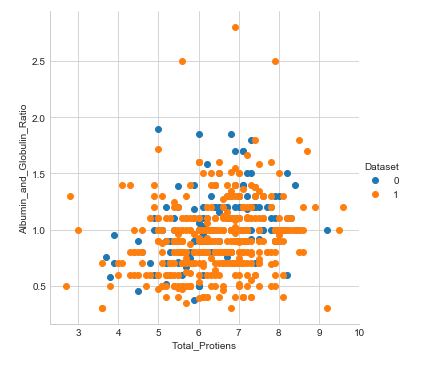
**** ****

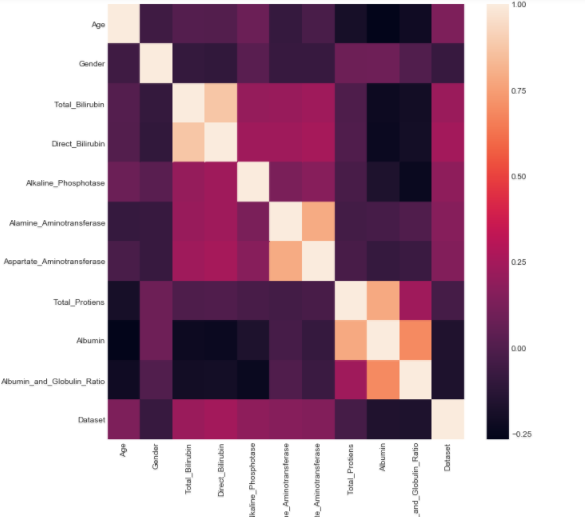
**count of gender:**

****

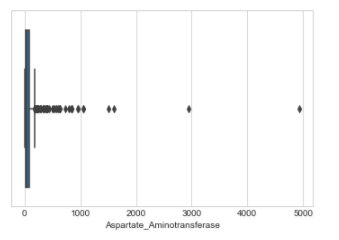
**Age seems to be a factor for liver disease:**

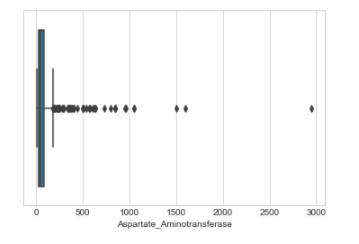
****

**   Features correlation:**



**Boxplot of Aspartate\_Aminotransferase:**



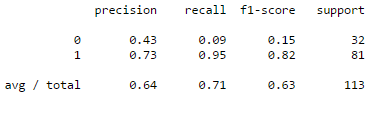


**Randomforest classifier:**

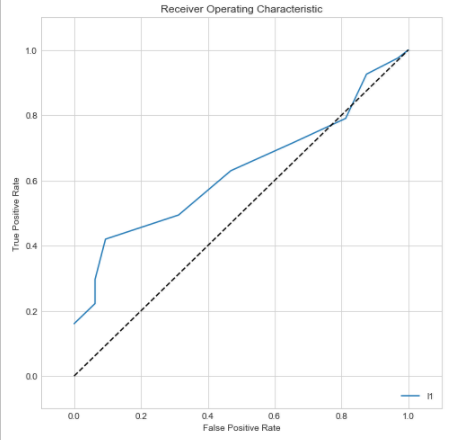
* Confusion matrix for the dataset:



* Accuracy of the model:70.79646017699115
* Classification report:



* AUC for train data:0.7782890007189072
* AUC for test data:0.6282793209876544
* ROC\_AUC curve:

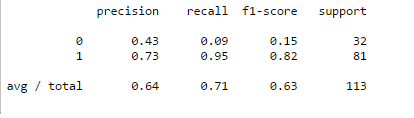


**SVM classifier:**

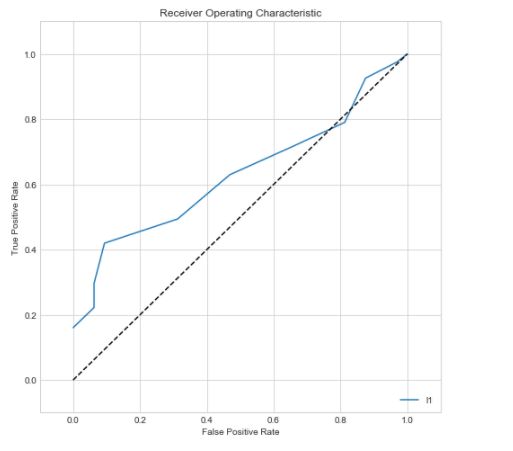
* Confusion matrix for the dataset:

****

* Accuracy of the model:70.79646017699115
* Classification report:

****

* AUC for train data:0.7782890007189072
* AUC for test data:0.6282793209876544
* ROC\_AUC curve:

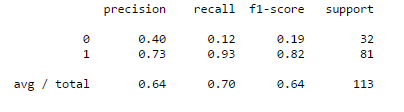
****

**KNN classifier:**

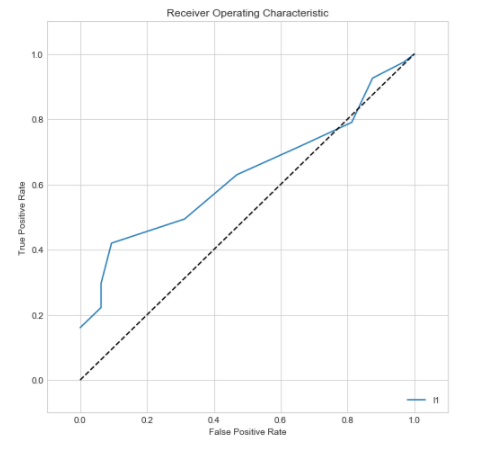
* Confusion matrix for the dataset:

****

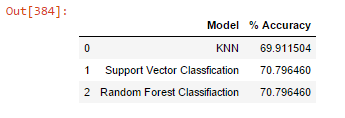
* Accuracy of the model:69.9115044247787
* Classification report:

****

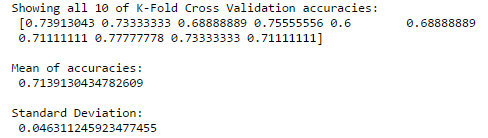
* AUC for train data:0.6376947040498442
* AUC for test data:0.6282793209876544
* ROC\_AUC curve:

****

**Model comparision:**



**Model evaluation:**

****

* The mean of accuracy score for the prediction from the test data was 0.7139130434782609
* We prepare the test prediction, from the averaged predictions for test over the 10 folds

1. **CONCLUSIONS:**

Liver disease prediction is a complex issue that requires a substantial amount of planning before throwing machine learning algorithms at it. Nonetheless, it is also an application of data science and machine learning for the good, which helps with doctors for their work.

Future work will include a comprehensive tuning of the Random Forest and SVM algorithms. Having a data set with non-anonymized features would make this particularly interesting as outputting the feature importance would enable one to see what specific factors are most important for detecting liver patient.

1. **REFERNCES:**

[1] https://www.medicinenet.com/liver\_disease/articl e.htm

[2] P. Rajeswari ,G. Sophia Reena , Analysis of Liver Disorder Using Data Mining Algorithm,Global Journal of Computer Science and Technology,2010.

[3] Sa’sdiyah Noor Novita Alfisahrin,Teddy Mantoro, Data mining Techniques For Optimatization of Liver Disease Classification,International conference on advanced Computer Science Application and Technologies,2013.

[4] S. Dhamodharan , Liver Disease Prediction Using Bayesian Classification , National Confrence on Advanced Computing,Application&Technologies,2014

[5] S.E.Sekar ,Y.Unal, Z.Erdem,and H.Erdinc Kocer,Ensembled Correlation Between Liver Analysis Output, International Journal of Biology and Biomedical ngineering,ISSN:1998-4150

[6] A.S.Aneesh kumar,Dr.C.Jothi Venkateswaran , A novel approach for Liver disorder Classification using Data Mining Techniques ,Engineering and Scientific International Journal ,ISSN 2394- 7179,ISSN 2394-7187,2015.

[7] P. Thangarajul,R.Mehala, Performance Analysis of PSO-KStar Classifier over Liver Diseases, International Journal of Advanced Research in Computer Engineering, 2015.

[8] Onwodi Gregory, Prediction of Liver Disease (Biliary Cirrhosis) Using Data Mining Technique, International Journal of EmergingTechnology&Research, ISSN (E):2347-5900, ISSN (P):2347-6079, 2015.

[9] Dr.S.Vijayarani,Mr.S.Dhayanand,Liver Disease Prediction using SVM and Naïve Bayes Algorithms, International Journal of Science, Engineering and Technology Research(IJSETR), 2015.

[10] Ebenezer Obaloluwa Olaniyi khashman Aadnan, “Liver DiseaseDiagnosisBasedon Neural Networks” , Advances in Computational Intelligence, Proceedings of the 16th International Conference on Neural Networks (NN '15), November 7-9, 2015.