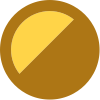
**Indian Liver Patient Records**

Patient records collected from North East of Andhra Pradesh, India



**Indian Liver Patient Records**



**About Dataset**

Context

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. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors.

Content

This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records.

Any patient whose age exceeded 89 is listed as being of age "90".

Columns:

* Age of the patient
* Gender of the patient
* Total Bilirubin
* Direct Bilirubin
* Alkaline Phosphotase
* Alamine Aminotransferase
* Aspartate Aminotransferase
* Total Protiens
* Albumin
* Albumin and Globulin Ratio
* Dataset: field used to split the data into two sets (patient with liver disease, or no disease)

*#Required Libraries Included in Development Environment*

import numpy as np

import pandas as pd

from pandas\_profiling import ProfileReport

*# Visiualization tools*

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

*#Machine Learning tools*

from sklearn.preprocessing import MinMaxScaler

*#Model Selection*

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

from sklearn.model\_selection import KFold

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

from sklearn.model\_selection import GridSearchCV

*#Machine Learning Models*

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import AdaBoostClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.tree import plot\_tree

*#Metrics*

from sklearn.metrics import accuracy\_score, confusion\_matrix, roc\_auc\_score, f1\_score

from yellowbrick.classifier import ROCAUC

In [2]:

*#import the dataset*

df = pd.read\_csv('../input/indian-liver-patient-records/indian\_liver\_patient.csv')

*#Replacing 1 with 0 (negative cases) and 2 with 1 (positive cases)*

df['Dataset'] = df['Dataset'].replace(1, 0)

df['Dataset'] = df['Dataset'].replace(2, 1)

*#print the 5 first rows of the dataset*

df.head()

Out[2]:

|  | Age | Gender | Total\_Bilirubin | Direct\_Bilirubin | Alkaline\_Phosphotase | Alamine\_Aminotransferase | Aspartate\_Aminotransferase | Total\_Protiens | Albumin | Albumin\_and\_Globulin\_Ratio | Dataset |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 65 | Female | 0.7 | 0.1 | 187 | 16 | 18 | 6.8 | 3.3 | 0.90 | 0 |
| 1 | 62 | Male | 10.9 | 5.5 | 699 | 64 | 100 | 7.5 | 3.2 | 0.74 | 0 |
| 2 | 62 | Male | 7.3 | 4.1 | 490 | 60 | 68 | 7.0 | 3.3 | 0.89 | 0 |
| 3 | 58 | Male | 1.0 | 0.4 | 182 | 14 | 20 | 6.8 | 3.4 | 1.00 | 0 |
| 4 | 72 | Male | 3.9 | 2.0 | 195 | 27 | 59 | 7.3 | 2.4 | 0.40 | 0 |

In [3]:

*#summary statistical information of numerical properties in the data set*

*#since the Gender feature contains non-numeric values, it has no statistical data*

df.describe().T

*#the range of the features are really different and this shows the need of normalization*

Out[3]:

|  | count | mean | std | min | 25% | 50% | 75% | max |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Age | 583.0 | 44.746141 | 16.189833 | 4.0 | 33.0 | 45.00 | 58.0 | 90.0 |
| Total\_Bilirubin | 583.0 | 3.298799 | 6.209522 | 0.4 | 0.8 | 1.00 | 2.6 | 75.0 |
| Direct\_Bilirubin | 583.0 | 1.486106 | 2.808498 | 0.1 | 0.2 | 0.30 | 1.3 | 19.7 |
| Alkaline\_Phosphotase | 583.0 | 290.576329 | 242.937989 | 63.0 | 175.5 | 208.00 | 298.0 | 2110.0 |
| Alamine\_Aminotransferase | 583.0 | 80.713551 | 182.620356 | 10.0 | 23.0 | 35.00 | 60.5 | 2000.0 |
| Aspartate\_Aminotransferase | 583.0 | 109.910806 | 288.918529 | 10.0 | 25.0 | 42.00 | 87.0 | 4929.0 |
| Total\_Protiens | 583.0 | 6.483190 | 1.085451 | 2.7 | 5.8 | 6.60 | 7.2 | 9.6 |
| Albumin | 583.0 | 3.141852 | 0.795519 | 0.9 | 2.6 | 3.10 | 3.8 | 5.5 |
| Albumin\_and\_Globulin\_Ratio | 579.0 | 0.947064 | 0.319592 | 0.3 | 0.7 | 0.93 | 1.1 | 2.8 |
| Dataset | 583.0 | 0.286449 | 0.452490 | 0.0 | 0.0 | 0.00 | 1.0 | 1.0 |

In [4]:

df.info()

*#the collumn Albumin\_and\_Globulin\_Ratio has 4 missing values and we have to purge this data to use the dataset in our model*

*#We need to transform gender into a numeric variable*

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 583 entries, 0 to 582

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Age 583 non-null int64

1 Gender 583 non-null object

2 Total\_Bilirubin 583 non-null float64

3 Direct\_Bilirubin 583 non-null float64

4 Alkaline\_Phosphotase 583 non-null int64

5 Alamine\_Aminotransferase 583 non-null int64

6 Aspartate\_Aminotransferase 583 non-null int64

7 Total\_Protiens 583 non-null float64

8 Albumin 583 non-null float64

9 Albumin\_and\_Globulin\_Ratio 579 non-null float64

10 Dataset 583 non-null int64

dtypes: float64(5), int64(5), object(1)

memory usage: 50.2+ KB

In [5]:

*#We can use some techniques to purge missing values. I choose to drop this rows*

df = df.dropna()

df.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 579 entries, 0 to 582

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Age 579 non-null int64

1 Gender 579 non-null object

2 Total\_Bilirubin 579 non-null float64

3 Direct\_Bilirubin 579 non-null float64

4 Alkaline\_Phosphotase 579 non-null int64

5 Alamine\_Aminotransferase 579 non-null int64

6 Aspartate\_Aminotransferase 579 non-null int64

7 Total\_Protiens 579 non-null float64

8 Albumin 579 non-null float64

9 Albumin\_and\_Globulin\_Ratio 579 non-null float64

10 Dataset 579 non-null int64

dtypes: float64(5), int64(5), object(1)

memory usage: 54.3+ KB