In [1]:

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

**first we need to load the heptitis dataset using pandas.**

In [2]:

data = pd.read\_csv('dataset\_hepatitis.csv')

**once data is loaded we need to check the shape/size and attributes in the dataset.**

In [3]:

data.columns

Out[3]:

Index(['AGE', 'SEX', 'STEROID', 'ANTIVIRALS', 'FATIGUE', 'MALAISE', 'ANOREXIA','LIVER\_BIG', 'LIVER\_FIRM', 'SPLEEN\_PALPABLE', 'SPIDERS', 'ASCITES',

'VARICES', 'BILIRUBIN', 'ALK\_PHOSPHATE', 'SGOT', 'ALBUMIN', 'PROTIME',

'HISTOLOGY', 'Class'],

dtype='object')

In [4]:

data.shape

Out[4]:

(155, 20)

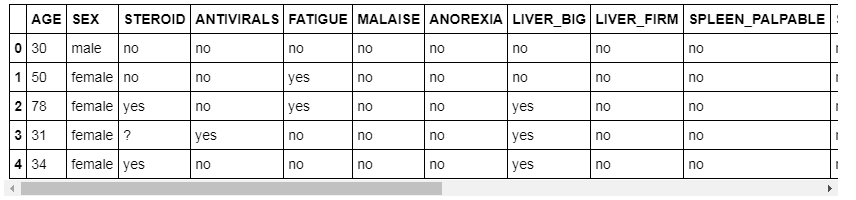
**now we need to check the values in the dataset and try to find out if we need to clean the data or not ?**

In [5]:

data.head()

*#data.tail()*

Out[5]:



**we have numerical as well as categorial data in the dataset.**

**we could see that most of the columns have missing data and its represented by "?"**

In [6]:

data['SEX'].value\_counts()

Out[6]:

female 139

male 16

Name: SEX, dtype: int64

In [7]:

data['Class'].value\_counts()

Out[7]:

LIVE 123

DIE 32

Name: Class, dtype: int64

**imbalanced dataset live class have 123 data's and die class have only 32 data's**

In [8]:

value = {'?':np.nan}

data.replace(value,inplace = **True**)

In [9]:

data['BILIRUBIN'] = data['BILIRUBIN'].astype(float)

data['ALK\_PHOSPHATE'] = data['ALK\_PHOSPHATE'].astype(float)

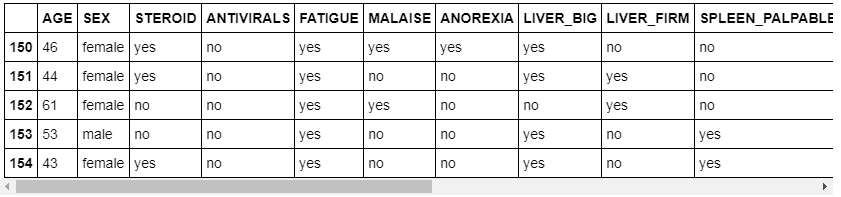
data['PROTIME'] = data['PROTIME'].astype(float)

data['SGOT'] = data['SGOT'].astype(float)

data['ALBUMIN'] = data['ALBUMIN'].astype(float)

data.tail()

Out[9]:



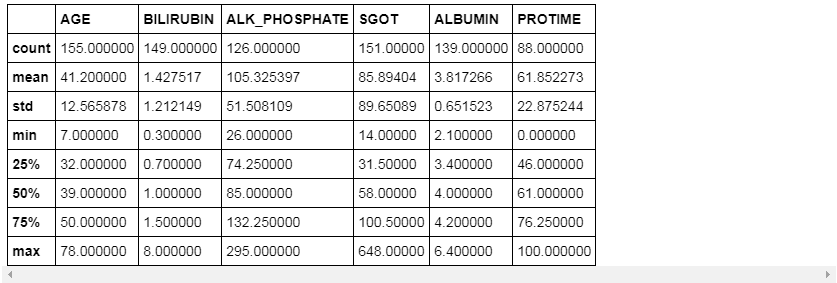
**"?" is replaced by NaN**

**int number are converted into float type to perform the describe()**

In [10]:

data.describe()

Out[10]:



**min age = 7 and max age = 78**

**we have 1 missing value in Bilirubin , 4 in SGOT, 29 in ALK\_PH, 67 in Protime**

**high std in ALK\_PHOSPHATE, SGOT, PROTIME**

In [11]:

data.median()

Out[11]:

AGE 39.0

BILIRUBIN 1.0

ALK\_PHOSPHATE 85.0

SGOT 58.0

ALBUMIN 4.0

PROTIME 61.0

dtype: float64

In [12]:

data.mean()

Out[12]:

AGE 41.200000

BILIRUBIN 1.427517

ALK\_PHOSPHATE 105.325397

SGOT 85.894040

ALBUMIN 3.817266

PROTIME 61.852273

dtype: float64

**ALK\_PHOSPHATE mean and median has very high difference.**

**To treat the missing value, we simply drop the NaN values !!**

**this method has lots of drawbacks as it reduces the sample size.**

In [13]:

new\_data = data.dropna()

In [14]:

new\_data.shape

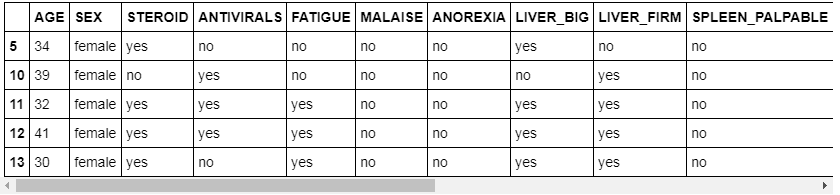
Out[14]:

(80, 20)

In [15]:

new\_data.head()

Out[15]:



**new create new-dataset and which contain 'BILIRUBIN','ALK\_PHOSPHATE','SGOT','ALBUMIN','PROTIME','Class'**

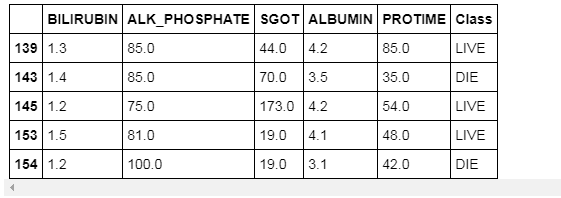
In [36]:

data2 = new\_data[['BILIRUBIN','ALK\_PHOSPHATE','SGOT','ALBUMIN','PROTIME','Class']]

In [37]:

data2.tail()

Out[37]:



In [38]:

plt.figure(1)

plt.subplot(121)

data2['BILIRUBIN'].hist()

plt.xlabel('BILIRUBIN')

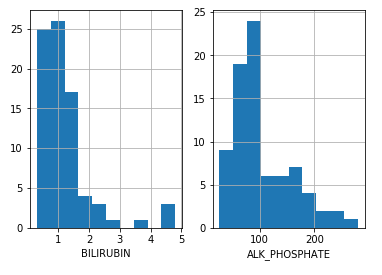
plt.subplot(122)

data2['ALK\_PHOSPHATE'].hist()

plt.xlabel('ALK\_PHOSPHATE')

Out[38]:

Text(0.5,0,'ALK\_PHOSPHATE')



In [39]:

plt.figure(1)

plt.subplot(121)

data2['SGOT'].hist()

plt.xlabel('SGOT')

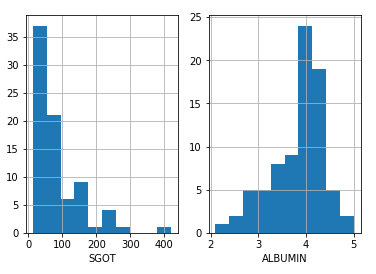
plt.subplot(122)

data2['ALBUMIN'].hist()

plt.xlabel('ALBUMIN')

Out[39]:

Text(0.5,0,'ALBUMIN')



In [40]:

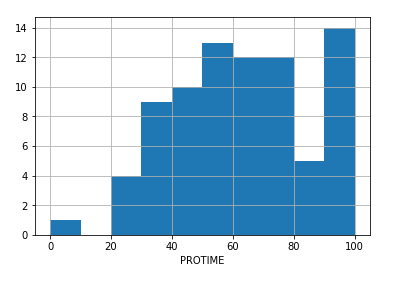
plt.figure(1)

data2['PROTIME'].hist()

plt.xlabel('PROTIME')

Out[40]:

Text(0.5,0,'PROTIME')



**From all the above histogram we found that all the plots have some kind of skewness.**

**To remove skewness we apply pandas function "applymap" and the numpy function "np.log" to logtransform the columns corresponding to those skewed variables in our dataframe.**

In [41]:

data2[['BILIRUBIN','ALK\_PHOSPHATE','SGOT','ALBUMIN']] = data2[['BILIRUBIN','ALK\_PHOSPHATE','SGOT','ALBUMIN']].applymap(np.log)

C:\Users\rpsir\Anaconda3\lib\site-packages\pandas\core\frame.py:3140: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

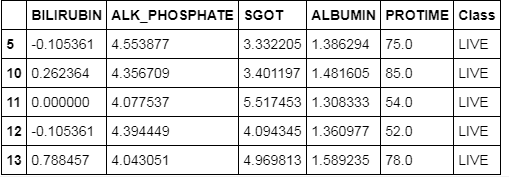
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy

self[k1] = value[k2]

In [42]:

data2.head()

Out[42]:



**since we have removed the skewness now we will use seaborn to use pair-plot function to visualize the relation btw the different numerical variables.**

In [45]:

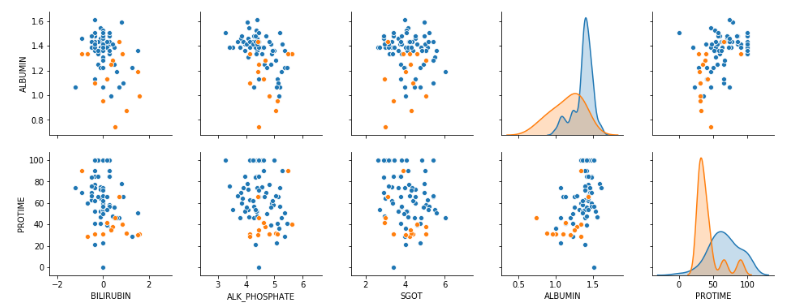
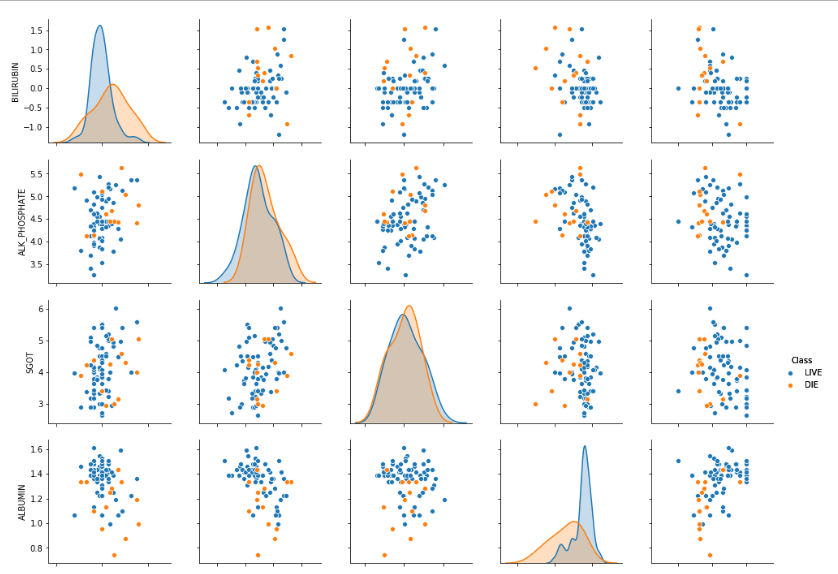
sns.pairplot(data2,hue = 'Class').add\_legend()

C:\Users\rpsir\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

return np.add.reduce(sorted[indexer] \* weights, axis=axis) / sumval

Out[45]:

<seaborn.axisgrid.PairGrid at 0xb5cf7f0>



**It appears that there is not a perfect linear relationship between the variables plotted**

In [ ]: