

# Using Neural Network algorithm to predict Hepatitis

February 4, 2021

```
[155]: #import the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
[156]: df=pd.read_csv('hepatitis.csv')
```

```
[157]: df.head()
```

```
[157]:
```

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	\
0	30	male	False	False	False	False	False	False	
1	50	female	False	False	True	False	False	False	
2	78	female	True	False	True	False	False	True	
3	31	female	NaN	True	False	False	False	True	
4	34	female	True	False	False	False	False	True	

	liver_firm	spleen_palpable	spiders	ascites	varices	bilirubin	\
0	False	False	False	False	False	1.0	
1	False	False	False	False	False	0.9	
2	False	False	False	False	False	0.7	
3	False	False	False	False	False	0.7	
4	False	False	False	False	False	1.0	

	alk_phosphate	sgot	albumin	prottime	histology	class
0	85.0	18.0	4.0	NaN	False	live
1	135.0	42.0	3.5	NaN	False	live
2	96.0	32.0	4.0	NaN	False	live
3	46.0	52.0	4.0	80.0	False	live
4	NaN	200.0	4.0	NaN	False	live

```
[158]: df['class'].replace({'live':1, 'die':0}, inplace=True)
```

```
[159]: df.isnull().sum()
```

```
[159]: age          0
sex            0
steroid        1
antivirals     0
```

```

fatigue          1
malaise          1
anorexia         1
liver_big       10
liver_firm      11
spleen_palpable  5
spiders          5
ascites          5
varices          5
bilirubin        6
alk_phosphate    29
sgot             4
albumin         16
protime         67
histology        0
class            0
dtype: int64

```

```
[160]: df.dtypes
```

```

[160]: age          int64
sex            object
steroid        object
antivirals      bool
fatigue        object
malaise        object
anorexia        object
liver_big      object
liver_firm      object
spleen_palpable object
spiders        object
ascites        object
varices        object
bilirubin      float64
alk_phosphate  float64
sgot           float64
albumin        float64
protime        float64
histology      bool
class          int64
dtype: object

```

```
[161]: df['steroid'].mode()
```

```

[161]: 0    True
dtype: object

```

```
[162]: df['steroid'].fillna(df['steroid'].mode().index[0], inplace=True)
```

```
[163]: df['steroid'].isna().sum()
```

```
[163]: 0
```

```
[164]: df['fatigue'].fillna(df['fatigue'].mode().index[0], inplace=True)
```

```
[165]: df['fatigue'].isna().sum()
```

```
[165]: 0
```

```
[166]: df['malaise'].fillna(df['malaise'].mode().index[0], inplace=True)
```

```
[167]: df['malaise'].isna().sum()
```

```
[167]: 0
```

```
[168]: df['anorexia'].fillna(df['anorexia'].mode().index[0], inplace=True)
```

```
[169]: df['anorexia'].isna().sum()
```

```
[169]: 0
```

```
[170]: df['liver_big'].fillna(df['liver_big'].mode().index[0], inplace=True)
```

```
[171]: df['liver_big'].isna().sum()
```

```
[171]: 0
```

```
[172]: df['liver_firm'].fillna(df['liver_firm'].mode().index[0], inplace=True)
```

```
[173]: df['spleen_palpable'].fillna(df['spleen_palpable'].mode().index[0],  
    ↪inplace=True)
```

```
[174]: df['spiders'].fillna(df['spiders'].mode().index[0], inplace=True)
```

```
[175]: df['ascites'].fillna(df['ascites'].mode().index[0], inplace=True)
```

```
[176]: df['varices'].fillna(df['varices'].mode().index[0], inplace=True)
```

```
[177]: df['bilirubin'].fillna(df['bilirubin'].mean(), inplace=True)
```

```
[178]: df['bilirubin'].isna().sum()
```

```
[178]: 0
```

```
[179]: df['bilirubin'].fillna(df['bilirubin'].mean(), inplace=True)
```

```
[180]: df['alk_phosphate'].describe()
#Here the descriptive statistics shows that there is difference between mean
↳and median so i prefer to go on imputation by median
```

```
[180]: count    126.000000
      mean     105.325397
      std      51.508109
      min      26.000000
      25%      74.250000
      50%      85.000000
      75%     132.250000
      max     295.000000
      Name: alk_phosphate, dtype: float64
```

```
[181]: df['alk_phosphate'].fillna(df['alk_phosphate'].median(), inplace=True)
```

```
[182]: df['alk_phosphate'].isna().sum()
```

```
[182]: 0
```

```
[183]: df['sgot'].describe()
```

```
[183]: count    151.00000
      mean      85.89404
      std      89.65089
      min      14.00000
      25%      31.50000
      50%      58.00000
      75%     100.50000
      max     648.00000
      Name: sgot, dtype: float64
```

```
[184]: df['sgot'].fillna(df['sgot'].median(), inplace=True)
```

```
[185]: df['sgot'].isna().sum()
```

```
[185]: 0
```

```
[186]: df['protime'].describe()
#There is no difference between mean and median, so i want to impute by mean
↳here
```

```
[186]: count    88.000000
      mean     61.852273
      std     22.875244
      min      0.000000
      25%     46.000000
      50%     61.000000
```

```
75%      76.250000
max      100.000000
Name: protime, dtype: float64
```

```
[187]: df['protime'].fillna(df['protime'].mean(), inplace=True)
```

```
[188]: df['protime'].isna().sum()
```

```
[188]: 0
```

```
[189]: df.isnull().sum()
```

```
[189]: age      0
sex      0
steroid   0
antivirals 0
fatigue   0
malaise   0
anorexia  0
liver_big 0
liver_firm 0
spleen_palpable 0
spiders   0
ascites   0
varices   0
bilirubin 0
alk_phosphate 0
sgot      0
albumin   16
protime   0
histology 0
class     0
dtype: int64
```

```
[190]: df['spleen_palpable'].fillna(df['spleen_palpable'].mode().index[0],
    →inplace=True)
```

```
[191]: df['spleen_palpable'].isna().sum()
```

```
[191]: 0
```

```
[192]: df['albumin'].describe()
```

```
[192]: count      139.000000
mean         3.817266
std          0.651523
min          2.100000
25%          3.400000
```

```

50%      4.000000
75%      4.200000
max       6.400000
Name: albumin, dtype: float64

```

```
[193]: df['albumin'].fillna(df['albumin'].mean(), inplace=True)
```

```
[194]: df['albumin'].isna().sum()
```

```
[194]: 0
```

```
[195]: df.isnull().sum().sum()
```

```
[195]: 0
```

```
[196]: df.head(n=10)
```

```
[196]:
```

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big \
0	30	male	False	False	False	False	False	False
1	50	female	False	False	True	False	False	False
2	78	female	True	False	True	False	False	True
3	31	female	0	True	False	False	False	True
4	34	female	True	False	False	False	False	True
5	34	female	True	False	False	False	False	True
6	51	female	False	False	True	False	True	True
7	23	female	True	False	False	False	False	True
8	39	female	True	False	True	False	False	True
9	30	female	True	False	False	False	False	True

	liver_firm	spleen_palpable	spiders	ascites	varices	bilirubin \
0	False	False	False	False	False	1.000000
1	False	False	False	False	False	0.900000
2	False	False	False	False	False	0.700000
3	False	False	False	False	False	0.700000
4	False	False	False	False	False	1.000000
5	False	False	False	False	False	0.900000
6	False	True	True	False	False	1.427517
7	False	False	False	False	False	1.000000
8	True	False	False	False	False	0.700000
9	False	False	False	False	False	1.000000

	alk_phosphate	sgot	albumin	prottime	histology	class
0	85.0	18.0	4.000000	61.852273	False	1
1	135.0	42.0	3.500000	61.852273	False	1
2	96.0	32.0	4.000000	61.852273	False	1
3	46.0	52.0	4.000000	80.000000	False	1
4	85.0	200.0	4.000000	61.852273	False	1

5	95.0	28.0	4.000000	75.000000	False	1
6	85.0	58.0	3.817266	61.852273	False	0
7	85.0	58.0	3.817266	61.852273	False	1
8	85.0	48.0	4.400000	61.852273	False	1
9	85.0	120.0	3.900000	61.852273	False	1

```
[197]: df['steroid'].value_counts()
```

```
[197]: True      78
      False    77
      Name: steroid, dtype: int64
```

```
[198]: df['steroid'].unique()
```

```
[198]: array([False,  True], dtype=object)
```

```
[199]: df1=df.copy()
```

```
[200]: df1.head(2)
```

```
[200]:   age    sex steroid  antivirals  fatigue  malaise  anorexia  liver_big \
0   30  male   False         False   False   False   False   False
1   50 female   False         False    True   False   False   False

      liver_firm  spleen_palpable  spiders  ascites  varices  bilirubin \
0         False              False   False   False   False         1.0
1         False              False   False   False   False         0.9

      alk_phosphate  sgot  albumin  protime  histology  class
0             85.0  18.0        4.0  61.852273     False     1
1            135.0  42.0        3.5  61.852273     False     1
```

```
[201]: df1['steroid']=pd.get_dummies(df1['steroid'])
```

```
[202]: df1['antivirals']=df1['antivirals'].astype(object)
```

```
[203]: df1['antivirals']=pd.get_dummies(df1['antivirals'])
```

```
[204]: df1.head(2)
```

```
[204]:   age    sex steroid  antivirals  fatigue  malaise  anorexia  liver_big \
0   30  male         1           1   False   False   False   False
1   50 female         1           1    True   False   False   False

      liver_firm  spleen_palpable  spiders  ascites  varices  bilirubin \
0         False              False   False   False   False         1.0
1         False              False   False   False   False         0.9
```

	alk_phosphate	sgot	albumin	prottime	histology	class
0	85.0	18.0	4.0	61.852273	False	1
1	135.0	42.0	3.5	61.852273	False	1

```
[205]: df1['histology']=df1['histology'].astype(object)
```

```
[206]: df1['histology']=pd.get_dummies(df1['histology'])
```

```
[207]: df1.head(2)
```

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big \
0	30	male	1	1	False	False	False	False
1	50	female	1	1	True	False	False	False

	liver_firm	spleen_palpable	spiders	ascites	varices	bilirubin \
0	False	False	False	False	False	1.0
1	False	False	False	False	False	0.9

	alk_phosphate	sgot	albumin	prottime	histology	class
0	85.0	18.0	4.0	61.852273	1	1
1	135.0	42.0	3.5	61.852273	1	1

```
[208]: df3=df1.copy()
```

```
[209]: df3.head(2)
```

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big \
0	30	male	1	1	False	False	False	False
1	50	female	1	1	True	False	False	False

	liver_firm	spleen_palpable	spiders	ascites	varices	bilirubin \
0	False	False	False	False	False	1.0
1	False	False	False	False	False	0.9

	alk_phosphate	sgot	albumin	prottime	histology	class
0	85.0	18.0	4.0	61.852273	1	1
1	135.0	42.0	3.5	61.852273	1	1

```
[210]: df3['fatigue']=pd.get_dummies(df3['fatigue'])
df3['malaise']=pd.get_dummies(df3['malaise'])
df3['anorexia']=pd.get_dummies(df3['anorexia'])
df3['liver_big']=pd.get_dummies(df3['liver_big'])
df3['liver_firm']=pd.get_dummies(df3['liver_firm'])
df3['spleen_palpable']=pd.get_dummies(df3['spleen_palpable'])
df3['spiders']=pd.get_dummies(df3['spiders'])
df3['ascites']=pd.get_dummies(df3['ascites'])
df3['varices']=pd.get_dummies(df3['varices'])
df3['sex']=pd.get_dummies(df3['sex'])
```



```
[211]: df3.head()
```

```
[211]:
```

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	\
0	30	0	1	1	1	1	1	1	
1	50	1	1	1	0	1	1	1	
2	78	1	0	1	0	1	1	0	
3	31	1	1	0	1	1	1	0	
4	34	1	0	1	1	1	1	0	

  

	liver_firm	spleen_palpable	spiders	ascites	varices	bilirubin	\
0	1	1	1	1	1	1.0	
1	1	1	1	1	1	0.9	
2	1	1	1	1	1	0.7	
3	1	1	1	1	1	0.7	
4	1	1	1	1	1	1.0	

  

	alk_phosphate	sgot	albumin	protime	histology	class
0	85.0	18.0	4.0	61.852273	1	1
1	135.0	42.0	3.5	61.852273	1	1
2	96.0	32.0	4.0	61.852273	1	1
3	46.0	52.0	4.0	80.000000	1	1
4	85.0	200.0	4.0	61.852273	1	1

```
[212]: df3.to_csv
```

```
[212]: <bound method NDFrame.to_csv of
```

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	\
0	30	0	1	1	1	1	1	1	
1	50	1	1	1	0	1	1	1	
2	78	1	0	1	0	1	1	0	
3	31	1	1	0	1	1	1	0	
4	34	1	0	1	1	1	1	0	
..	...	...	...	...	...	...	...	...	
150	46	1	0	1	0	0	0	0	
151	44	1	0	1	0	1	1	0	
152	61	1	1	1	0	0	1	1	
153	53	0	1	1	0	1	1	0	
154	43	1	0	1	0	1	1	0	

  

	liver_firm	spleen_palpable	spiders	ascites	varices	bilirubin	\
0	1	1	1	1	1	1.0	
1	1	1	1	1	1	0.9	
2	1	1	1	1	1	0.7	
3	1	1	1	1	1	0.7	
4	1	1	1	1	1	1.0	
..	...	...	...	...	...	...	
150	1	1	0	0	0	7.6	

151	0	1	1	1	1	0.9
152	0	1	0	1	1	0.8
153	1	0	0	1	0	1.5
154	1	0	0	0	1	1.2

	alk_phosphate	sgot	albumin	prottime	histology	class
0	85.0	18.0	4.0	61.852273	1	1
1	135.0	42.0	3.5	61.852273	1	1
2	96.0	32.0	4.0	61.852273	1	1
3	46.0	52.0	4.0	80.000000	1	1
4	85.0	200.0	4.0	61.852273	1	1
..	...	...	...	...	...	...
150	85.0	242.0	3.3	50.000000	0	0
151	126.0	142.0	4.3	61.852273	0	1
152	75.0	20.0	4.1	61.852273	0	1
153	81.0	19.0	4.1	48.000000	0	1
154	100.0	19.0	3.1	42.000000	0	0

[155 rows x 20 columns]>

```
[213]: import os
```

```
[214]: pwd
```

```
[214]: 'C:\\Users\\sudip\\Desktop\\Hepataitis'
```

```
[215]: df3.to_csv('C:\\Users\\sudip\\Desktop\\Hepataitis.csv')
```

```
[216]: #split the dataframe into x and y variables
x=df3.drop('class', axis=1).values
```

```
[217]: y=df3['class'].values
```

```
[218]: #build a Machine Learning model
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.25,
↪random_state=62)
```

```
[219]: from sklearn.linear_model import LogisticRegression
```

```
[220]: lr=LogisticRegression(solver='lbfgs')
```

```
[221]: lr.fit(x_train, y_train)
```

```
[221]: LogisticRegression()
```

```
[222]: y_pred=lr.predict(x_test)
```

```
[223]: y_pred
```

```
[223]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
[224]: accuracy_score(y_test, y_pred)
```

```
[224]: 0.8974358974358975
```

```
[225]: from sklearn.metrics import confusion_matrix, classification_report
```

```
[226]: confusion_matrix(y_test, y_pred)
```

```
[226]: array([[ 2,  2],
        [ 2, 33]], dtype=int64)
```

```
[227]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.50	0.50	0.50	4
1	0.94	0.94	0.94	35
accuracy			0.90	39
macro avg	0.72	0.72	0.72	39
weighted avg	0.90	0.90	0.90	39

```
[228]: from sklearn.neural_network import MLPClassifier
```

```
[245]: classifier= MLPClassifier(hidden_layer_sizes=(100,100), activation='tanh',  
    ↪max_iter=40)
```

```
[246]: classifier.fit(x_train, y_train)
```

```
[246]: MLPClassifier(activation='tanh', hidden_layer_sizes=(100, 100), max_iter=40)
```

```
[247]: y_pred1_nn= classifier.predict(x_test)
```

```
[248]: y_pred1_nn
```

```
[248]: array([0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
[249]: confusion_matrix(y_test, y_pred1_nn)
```

```
[249]: array([[ 1,  3],  
            [ 7, 28]], dtype=int64)
```

```
[250]: from sklearn.metrics import accuracy_score
```

```
[251]: accuracy_score(y_test, y_pred1_nn)
```

```
[251]: 0.7435897435897436
```

```
[252]: print(classification_report(y_test, y_pred1_nn))
```

	precision	recall	f1-score	support
0	0.12	0.25	0.17	4
1	0.90	0.80	0.85	35
accuracy			0.74	39
macro avg	0.51	0.53	0.51	39
weighted avg	0.82	0.74	0.78	39