```
In [1]: import pandas as pd
         import numpy as np
In [2]: path = 'https://raw.githubusercontent.com/ovibaridar/Data_sets/main/hepatitis_2.csv'
In [3]: data = pd.read_csv(path)
In [4]: data.head()
Out[4]:
                                                                               LIVER LIVER
                                                                                              SPLEEN
            Class AGE SEX STEROID ANTIVIRALS FATIGUE MALAISE ANOREXIA
                                                                                                       SPIDERS ASCITES VARICES BILIRUBIN
                                                                                      FIRM
                                                                                            PALPABLE
         0
                                                        2
                          2
                                               2
                                                                 2
                                                                            2
                                                                                  1.0
                                                                                                                                        1.0
                0
                    30
                                  1.0
                                                                                        2.0
                                                                                                   2.0
                                                                                                            2.0
                                                                                                                     2.0
                                                                                                                              2.0
                0
                    50
                          1
                                  1.0
                                               2
                                                        1
                                                                 2
                                                                            2
                                                                                  1.0
                                                                                        2.0
                                                                                                   2.0
                                                                                                            2.0
                                                                                                                     2.0
                                                                                                                              2.0
                                                                                                                                        0.9
                0
                    78
                                  2.0
                                               2
                                                                  2
                                                                            2
                                                                                  2.0
                                                                                        2.0
                                                                                                   2.0
                                                                                                            2.0
                                                                                                                              2.0
                                                                                                                                        0.7
                0
                                                        2
                                                                 2
                                                                            2
                                                                                        2.0
                                                                                                   2.0
                                                                                                            2.0
                                                                                                                     2.0
                                                                                                                              2.0
                    31
                          1
                                 NaN
                                                                                  2.0
                                                                                                                                        0.7
                                               2
                                                        2
                0
                    34
                          1
                                  2.0
                                                                 2
                                                                            2
                                                                                  2.0
                                                                                        2.0
                                                                                                   2.0
                                                                                                            2.0
                                                                                                                     2.0
                                                                                                                              2.0
                                                                                                                                        1.0
In [5]: data.isnull().sum()
Out[5]: Class
         AGE
                               0
         SEX
         STEROID
                              1
         ANTIVIRALS
                              0
         FATIGUE
                               0
         MALAISE
                              0
         ANOREXIA
                              0
         LIVER BIG
         LIVER FIRM
                             10
         SPLEEN PALPABLE
         SPIDERS
                              4
         ASCITES
                              4
         VARICES
         BILIRUBIN
                              5
         ALK PHOSPHATE
                             28
         SGOT
                              3
         ALBUMIN
                             15
         PROTIME
                             66
         HISTOLOGY
         dtype: int64
```

Remove Null Values from Data

```
In [7]: data.isnull().sum()
Out[7]: Class
                            0
        AGE
        SEX
                            0
        STEROID
                            0
        ANTIVIRALS
        FATIGUE
                            0
        MALAISE
                            0
        ANOREXIA
                            0
        LIVER BIG
                            0
        LIVER FIRM
                            0
        SPLEEN PALPABLE
                            0
        SPIDERS
        ASCITES
                            0
        VARICES
                            0
        BILIRUBIN
                            0
        ALK PHOSPHATE
                            0
                            0
        SGOT
        ALBUMIN
                            0
        PROTIME
                            0
        HISTOLOGY
                            0
        dtype: int64
In [8]: data.tail(10)
Out[8]:
                                                                               LIVER
                                                                                        LIVER
                                                                                                SPLEEN
              Class AGE SEX STEROID ANTIVIRALS FATIGUE MALAISE ANOREXIA
                                                                                              PALPABLE
                                                                                        FIRM
                                                                                 BIG
```

SPIDERS ASCITES VARICES BIL 144 0 31 2 2 2 2.000000 2.000000 2.000000 2.00 2 145 41 2.0 2 2 2.000000 1.0 2.000000 1.00 1 1 1.000000 1.00 146 1 70 1 1.0 2 1 1 1.827586 1.583333 1.8 1.66 1.866667 1.88 147 0 20 1.0 2 2 2 2.000000 1.583333 2.0 2.00 2.000000 2.00 148 0 36 2.0 2 2 2.000000 2.000000 2.0 2.000000 2.00 2.00 2 1.000000 149 1 46 1 2.0 1 1 2.000000 2.000000 2.0 1.00 1.00 0 2 2 2.0 2.00 150 44 2.0 2.000000 1.000000 2.00 2.000000 151 0 61 1.0 2 1.000000 1.000000 2.0 1.00 2.000000 2.00 0 53 2 2 2 1.0 2.000000 1.00 152 1.0 1 2.000000 2.000000 1.00 2 153 1 43 1 2 2 2.000000 1.000000 2.00 2.0 2.000000 1.0 1.00

In [9]: data.head()

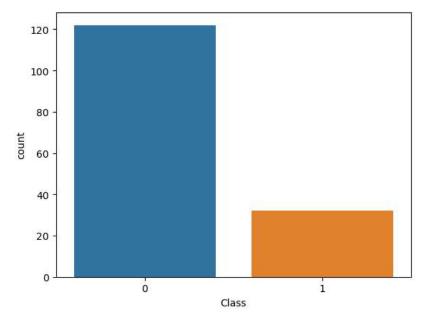
Out[9]:

	Class	AGE	SEX	STEROID	ANTIVIRALS	FATIGUE	MALAISE	ANOREXIA	LIVER BIG	LIVER FIRM	SPLEEN PALPABLE	SPIDERS	ASCITES	VARICES	BILIRUBIN
0	0	30	2	1.000000	2	2	2	2	1.0	2.0	2.0	2.0	2.0	2.0	1.0
1	0	50	1	1.000000	2	1	2	2	1.0	2.0	2.0	2.0	2.0	2.0	0.9
2	0	78	1	2.000000	2	1	2	2	2.0	2.0	2.0	2.0	2.0	2.0	0.7
3	0	31	1	1.509804	1	2	2	2	2.0	2.0	2.0	2.0	2.0	2.0	0.7
4	0	34	1	2.000000	2	2	2	2	2.0	2.0	2.0	2.0	2.0	2.0	1.0
4															+

Graphical visualization

In [57]: import seaborn as sns

```
In [58]: sns.countplot(x= 'Class', data = data )
Out[58]: <Axes: xlabel='Class', ylabel='count'>
```



Split the dataset into features (X) and target variable (y), where 1 indicates a positive case and 0 indicates a negative case.

```
In [59]: x = data.drop('Class' , axis=1)
y = data[['Class']]
In [60]: x.head()
Out[60]:
```

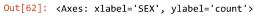
	AGE	SEX	STEROID	ANTIVIRALS	FATIGUE	MALAISE	ANOREXIA	LIVER BIG	LIVER FIRM	SPLEEN PALPABLE	SPIDERS	ASCITES	VARICES	BILIRUBIN	PHOS
0	30	2	1.000000	2	2	2	2	1.0	2.0	2.0	2.0	2.0	2.0	1.0	85
1	50	1	1.000000	2	1	2	2	1.0	2.0	2.0	2.0	2.0	2.0	0.9	135
2	78	1	2.000000	2	1	2	2	2.0	2.0	2.0	2.0	2.0	2.0	0.7	96
3	31	1	1.509804	1	2	2	2	2.0	2.0	2.0	2.0	2.0	2.0	0.7	46
4	34	1	2.000000	2	2	2	2	2.0	2.0	2.0	2.0	2.0	2.0	1.0	105
4															>

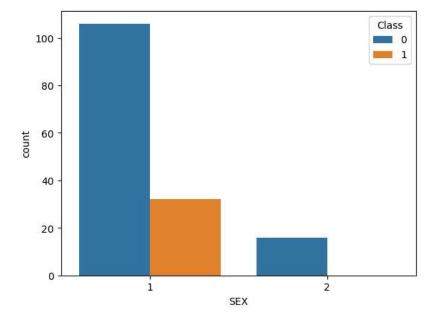
```
In [61]: sns.countplot(x='AGE' , data=data, hue='Class')
Out[61]: <Axes: xlabel='AGE', ylabel='count'>
```

7-6-5-5-3-2-1-

72**222222333333333333434343434355533465356556565977**8 AGE

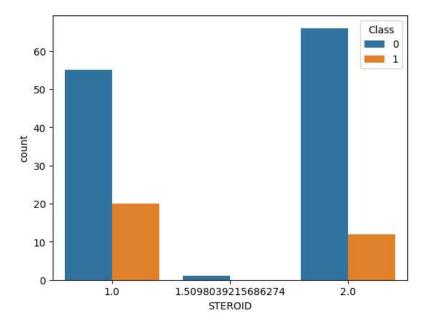
```
In [62]: sns.countplot(x='SEX' , data=data, hue='Class')
```





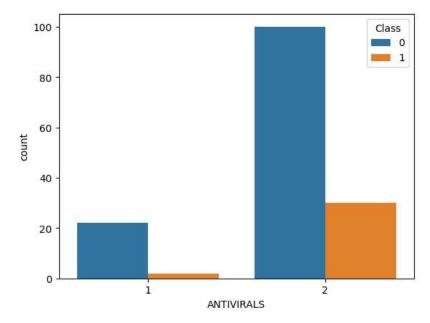
```
In [63]: sns.countplot(x='STEROID' , data=data, hue='Class')
```

Out[63]: <Axes: xlabel='STEROID', ylabel='count'>



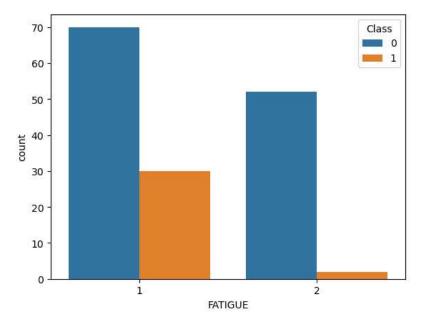
```
In [64]: sns.countplot(x='ANTIVIRALS' , data=data, hue='Class')
```

Out[64]: <Axes: xlabel='ANTIVIRALS', ylabel='count'>



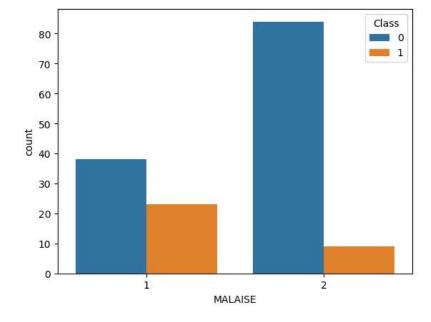
```
In [65]: sns.countplot(x='FATIGUE' , data=data, hue='Class')
```

Out[65]: <Axes: xlabel='FATIGUE', ylabel='count'>



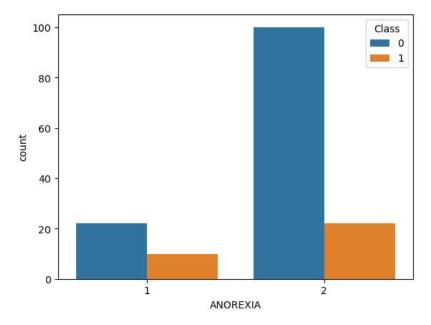
```
In [66]: sns.countplot(x='MALAISE' , data=data, hue='Class')
```

Out[66]: <Axes: xlabel='MALAISE', ylabel='count'>



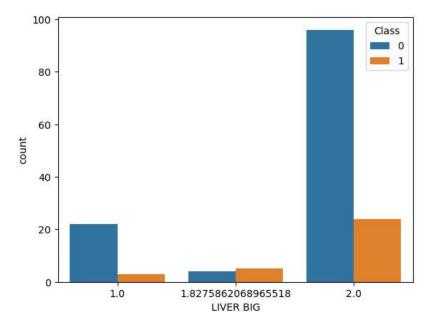
```
In [67]: sns.countplot(x='ANOREXIA' , data=data, hue='Class')
```

Out[67]: <Axes: xlabel='ANOREXIA', ylabel='count'>



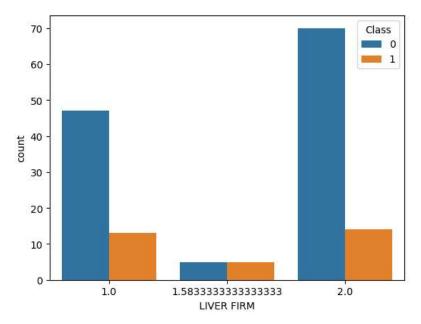
In [68]: sns.countplot(x='LIVER BIG' , data=data, hue='Class')

Out[68]: <Axes: xlabel='LIVER BIG', ylabel='count'>



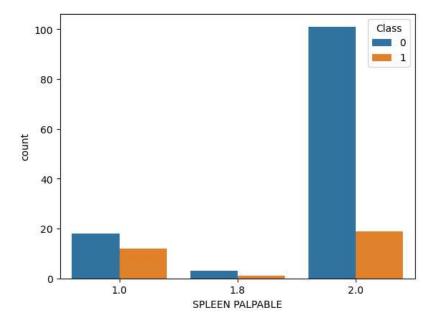
```
In [69]: sns.countplot(x='LIVER FIRM', data=data, hue='Class')
```

Out[69]: <Axes: xlabel='LIVER FIRM', ylabel='count'>



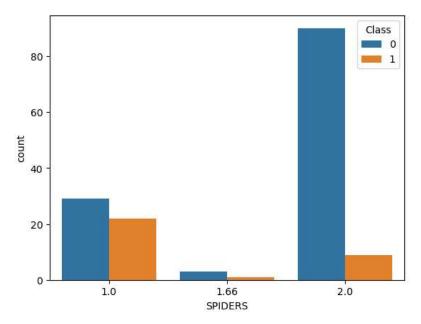
```
In [70]: sns.countplot(x='SPLEEN PALPABLE' , data=data, hue='Class')
```

Out[70]: <Axes: xlabel='SPLEEN PALPABLE', ylabel='count'>



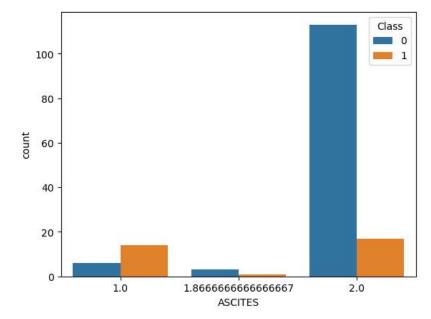
```
In [71]: sns.countplot(x='SPIDERS' , data=data, hue='Class')
```

Out[71]: <Axes: xlabel='SPIDERS', ylabel='count'>



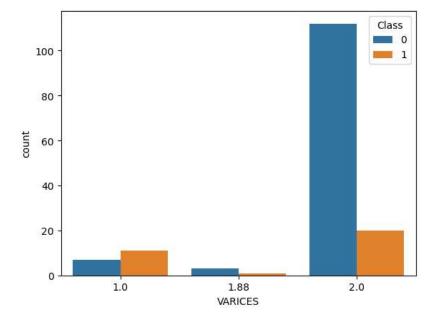
```
In [72]: sns.countplot(x='ASCITES' , data=data, hue='Class')
```

Out[72]: <Axes: xlabel='ASCITES', ylabel='count'>



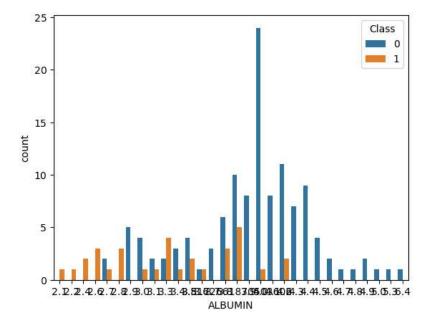
```
In [73]:
sns.countplot(x='VARICES' , data=data, hue='Class')
```

Out[73]: <Axes: xlabel='VARICES', ylabel='count'>

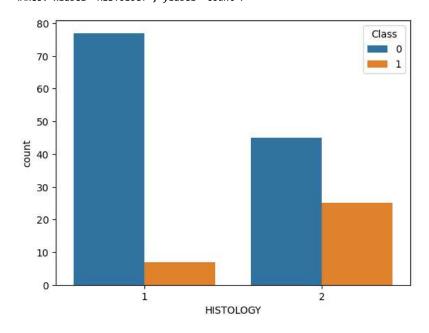


```
In [74]: sns.countplot(x='ALBUMIN' , data=data, hue='Class')
```

Out[74]: <Axes: xlabel='ALBUMIN', ylabel='count'>



```
In [101]: sns.countplot(x='HISTOLOGY' , data=data, hue='Class')
Out[101]: <Axes: xlabel='HISTOLOGY', ylabel='count'>
```



Split the dataset into training (70%) and testing sets.

```
In [ ]: from sklearn.model_selection import train_test_split

In [174]: xtrain,xtest,ytrain,ytest = train_test_split(x,y,random_state=40 , test_size=.25)

In [175]: xtrain.shape

Out[175]: (115, 19)

In [176]: xtest.shape

Out[176]: (39, 19)
```

```
In [177]: sns.heatmap(xtrain.corr(), annot=True)
Out[177]: <Axes: >
                                                                                                                -1.0
                               AGE - 1, 00:530,00:0926. D3007002694.D30.2.06.D30546.030.40.20.140.2
                               SEX -00 1 .04.09 903 50 00 00 00 00 00 10 040 5203 40 00 3050 10342
                          STEROID -0.60704 1 .0208150.0.040826.102.00.6302.0040.670080-86056620.10.1
                                                                                                                - 0.8
                      ANTIVIRALS -001.9392 1 .0093059. D.101.060816.29.19.10819.20.1-8.10.1 D.2
                          FATIGUE -0.280 050509 1 0.68.390 8042 9.26.40.28.20.29.160.20.30.24.1
                          MALAISE -0. D3010-2.0595 10.538080404010339.39.1-6.25.20.25.28.28.06
                                                                                                               - 0.6
                        ANOREXIA -- 00.000.048. D.39.56 1 D.10200.056.30.20.1-8.29.10.20.160-9803
                        LIVER BIG -.02610526.0006498041211.0420709180-0.00.6107.004040840105.00
                                                                                                                -0.4
                       LIVER FIRM -. 94.0 06020 602 4. 1040 7847 1 0.26-8 20 09.308 0 705-3030 8/51 6.04.1
               SPLEEN PALPABLE -.06.06.0-0610.20600.30600.792 10.301040.207.0800200054840-55.1
                          SPIDERS -0.02.00.90302.204.40.3.40.30.16.30.31 1 0.39.48.40.2060.843.6.2-9.3
                                                                                                                - 0.2
                          ASCITES -. 0 671-9 . 09. 109 2 8 . 3 9 . 2 29010 09:0940 3 9 1 0 . 3 8 . 3050 9 . 30 1035 9 . 2-9 . 3
                          VARICES -.036-520-6710820.16.1080 D616.20.46.3 1 0.46.2090 20329.24.3
                                                                                                                - 0.0
                        BILIRUBIN -0.105.0 3340 6.1-9.240.25.290 0 70 0 50 4034 2.35.4 1 0.10.290 -30.205.1
                 ALK PHOSPHATE -.03800.030528.16.20.10.09.330.20.206093291 1 1.19.30.10.2
                              SGOT -0.-D.00350506180.-20.25.28043080905.48.490.3020329.18 1 D.14.13.2
                                                                                                                - -0.2
                         ALBUMIN -3020000180.19.30.28.16.10.10608936.50.290.30.30.1 10.28.2
                         PROTIME -0. 1040 304 10. 10. 20. 2080 980 1010040 505 20. 20. 20. 20. 26. 105. 10. 28 1 0. 2
                       HISTOLOGY -0.20.20.14.20.1006.903040816.10.34.30.30.19.240.20.26.2 1
                                                                                                                  -0.4
                                                   FATIGUE –
MALAISE –
ANOREXIA –
LIVER BIG –
LIVER FIRM –
SPLEEN PALPABLE –
                                                                        SPIDERS -
ASCITES -
VARICES -
                                             STEROID
                                                                                   BILIRUBIN
                                                                                       ALK PHOSPHATE
```

- Build a Decision Tree classifier using a suitable library (scikit-learn).

```
In [178]: from sklearn.tree import DecisionTreeClassifier
import warnings
warnings.filterwarnings("ignore")

In [179]: dtc = DecisionTreeClassifier()
```

- Train the model on the training set.

- Make predictions on the testing set.

```
In [183]: perd= dtc.predict(xtest)
perd

Out[183]: array([1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

Out[190]: 0.6

```
In [184]: from sklearn.metrics import confusion_matrix ,precision_score ,recall_score ,accuracy_score , f1_score , RocCurveDispla
In [185]: cm = confusion_matrix(ytest, perd)
Out[185]: array([[25, 5],
                 [ 3, 6]], dtype=int64)
In [186]: sns.heatmap(cm , annot=True)
Out[186]: <Axes: >
                                                                        - 25.0
                                                                         - 22.5
                                                                        - 20.0
           0 -
                          25
                                                                        - 17.5
                                                                        - 15.0
                                                                        - 12.5
                                                                        - 10.0
                                                                         - 7.5
                                                                         - 5.0
                           ó
                                                     i
In [187]: precision_score(ytest,perd)
Out[187]: 0.54545454545454
In [188]: recall_score(ytest,perd)
Out[188]: 0.66666666666666
In [189]: accuracy_score(ytest,perd)
Out[189]: 0.7948717948717948
In [190]: f1_score(ytest,perd)
```

```
In [191]: from matplotlib import pyplot as plt
RocCurveDisplay.from_predictions(ytest,perd)
plt.plot([0,1],[0,1])
```

Out[191]: [<matplotlib.lines.Line2D at 0x251cd06ad90>]

