EVAL LAB 5

Zeel Mehta 21BAI1533

In [262]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [263]:
```
df=pd.read_csv('/Users/zeelmehta/Desktop/Indian Liver Patient Datas
```

In [264]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 582 entries, 0 to 581
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   65      582 non-null    int64  
 1   Female  582 non-null    object 
 2   0.7     582 non-null    float64
 3   0.1     582 non-null    float64
 4   187     582 non-null    int64  
 5   16      582 non-null    int64  
 6   18      582 non-null    int64  
 7   6.8     582 non-null    float64
 8   3.3     582 non-null    float64
 9   0.9     578 non-null    float64
 10  1       582 non-null    int64  
dtypes: float64(5), int64(5), object(1)
memory usage: 50.1+ KB
```

In [265]:
```
df.isnull().sum()
```

Out[265]:
```
65        0
Female    0
0.7       0
0.1       0
187       0
16        0
18        0
6.8       0
3.3       0
0.9       4
1         0
dtype: int64
```

In [266]:
```
df.columns
```

Out[266]:
```
Index(['65', 'Female', '0.7', '0.1', '187', '16', '18', '6.8', '3.
3', '0.9',
       '1'],
      dtype='object')
```

In [267]:
```
df.dtypes
```

Out[267]:
```
65          int64
Female     object
0.7       float64
0.1       float64
187         int64
16          int64
18          int64
6.8       float64
3.3       float64
0.9       float64
1           int64
dtype: object
```

In [268]:
```
df.head()
```

Out[268]:

	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.9	1
0	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
1	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
2	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
3	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1
4	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	1

Perform pre-processing if required (2 Marks)

In [269]:
```
df.columns =['age', 'gender', 'tb', 'db','alkphos','sgpt','sgot','t
df.head()
```

Out[269]:

	age	gender	tb	db	alkphos	sgpt	sgot	tp	alb	agratio	label
0	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
1	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
2	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
3	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1
4	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	1

In [270]:
```
from sklearn.preprocessing import LabelEncoder
def Encoder(df):
          col=df['gender']
          le = LabelEncoder()
          for feature in col:
              try:
                  df['gender'] = le.fit_transform(df['gender'])
              except:
                  print('Error encoding '+feature)
          return df
```

In [271]:
```
Encoder(df)
df['gender']
```

Out[271]:
```
0      1
1      1
2      1
3      1
4      1
      ..
577    1
578    1
579    1
580    1
581    1
Name: gender, Length: 582, dtype: int64
```

In [272]:
```
df.isnull().sum()
```

Out[272]:
```
age         0
gender      0
tb          0
db          0
alkphos     0
sgpt        0
sgot        0
tp          0
alb         0
agratio     4
label       0
dtype: int64
```

In [273]:
```
from sklearn.impute import SimpleImputer
imp=SimpleImputer(missing_values=np.NaN,strategy='mean')
df['agratio']=imp.fit_transform(x)
```

In [274]:
```
df['agratio'].isnull().sum()
```

Out[274]:
```
0
```

In [275]:
```
df.head()
```

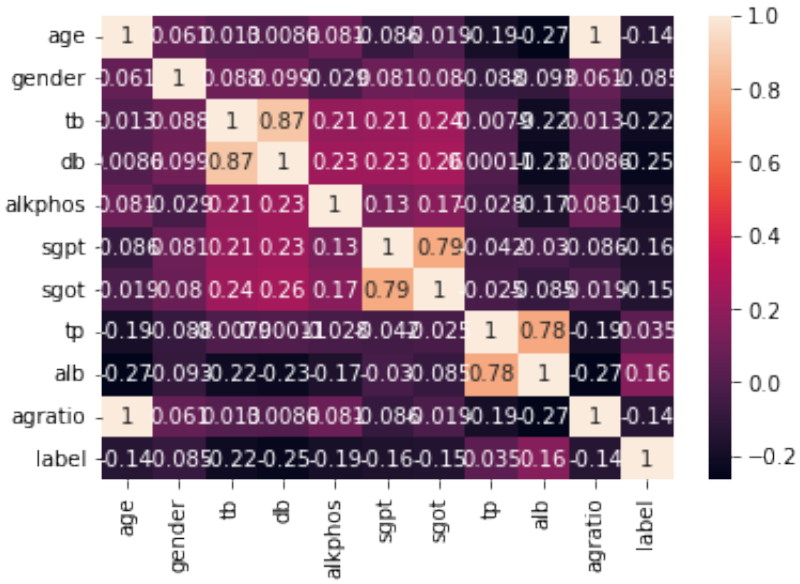
Out[275]:

	age	gender	tb	db	alkphos	sgpt	sgot	tp	alb	agratio	label
0	62	1	10.9	5.5	699	64	100	7.5	3.2	62.0	1
1	62	1	7.3	4.1	490	60	68	7.0	3.3	62.0	1
2	58	1	1.0	0.4	182	14	20	6.8	3.4	58.0	1
3	72	1	3.9	2.0	195	27	59	7.3	2.4	72.0	1
4	46	1	1.8	0.7	208	19	14	7.6	4.4	46.0	1

Draw the heat map (2 Marks)

In [276]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[276]: <AxesSubplot:>



Split the dataset into Train and Test.

In [277]:
```
from sklearn.model_selection import train_test_split
```

In [278]:
```
x=df.drop('label',axis=1)
x
```

Out[278]:

	age	gender	tb	db	alkphos	sgpt	sgot	tp	alb	agratio
0	62	1	10.9	5.5	699	64	100	7.5	3.2	62.0
1	62	1	7.3	4.1	490	60	68	7.0	3.3	62.0
2	58	1	1.0	0.4	182	14	20	6.8	3.4	58.0
3	72	1	3.9	2.0	195	27	59	7.3	2.4	72.0
4	46	1	1.8	0.7	208	19	14	7.6	4.4	46.0
...	...	...	...	...	...	...	...	...	...	...
577	60	1	0.5	0.1	500	20	34	5.9	1.6	60.0
578	40	1	0.6	0.1	98	35	31	6.0	3.2	40.0
579	52	1	0.8	0.2	245	48	49	6.4	3.2	52.0
580	31	1	1.3	0.5	184	29	32	6.8	3.4	31.0
581	38	1	1.0	0.3	216	21	24	7.3	4.4	38.0

582 rows × 10 columns

In [279]:
```
y=df['label']
y
```

Out[279]:
```
0      1
1      1
2      1
3      1
4      1
      ..
577    2
578    1
579    1
580    1
581    2
Name: label, Length: 582, dtype: int64
```

In [280]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [281]:
```
x_train.head()
```

Out[281]:

	age	gender	tb	db	alkphos	sgpt	sgot	tp	alb	agratio
149	56	1	1.1	0.5	180	30	42	6.9	3.8	56.0
470	49	0	0.6	0.1	185	17	26	6.6	2.9	49.0
498	55	1	3.6	1.6	349	40	70	7.2	2.9	55.0
404	31	0	0.8	0.2	215	15	21	7.6	4.0	31.0
483	62	1	5.0	2.1	103	18	40	5.0	2.1	62.0

In [282]:
```
x_test.head()
```

Out[282]:

	age	gender	tb	db	alkphos	sgpt	sgot	tp	alb	agratio
254	38	1	1.7	0.7	859	89	48	6.0	3.0	38.0
321	36	1	1.7	0.5	205	36	34	7.1	3.9	36.0
258	42	1	30.5	14.2	285	65	130	5.2	2.1	42.0
83	32	1	0.6	0.1	237	45	31	7.5	4.3	32.0
409	18	1	1.4	0.6	215	440	850	5.0	1.9	18.0

In [283]:
```
y_train.head()
```

Out[283]:
```
149    2
470    2
498    1
404    1
483    1
Name: label, dtype: int64
```

In [284]:
```
y_test.head()
```

Out[284]:
```
254    1
321    1
258    1
83     1
409    1
Name: label, dtype: int64
```

Apply Decision Tree on the data(3 Marks)

In [285]:
```
from sklearn.tree import DecisionTreeClassifier
```

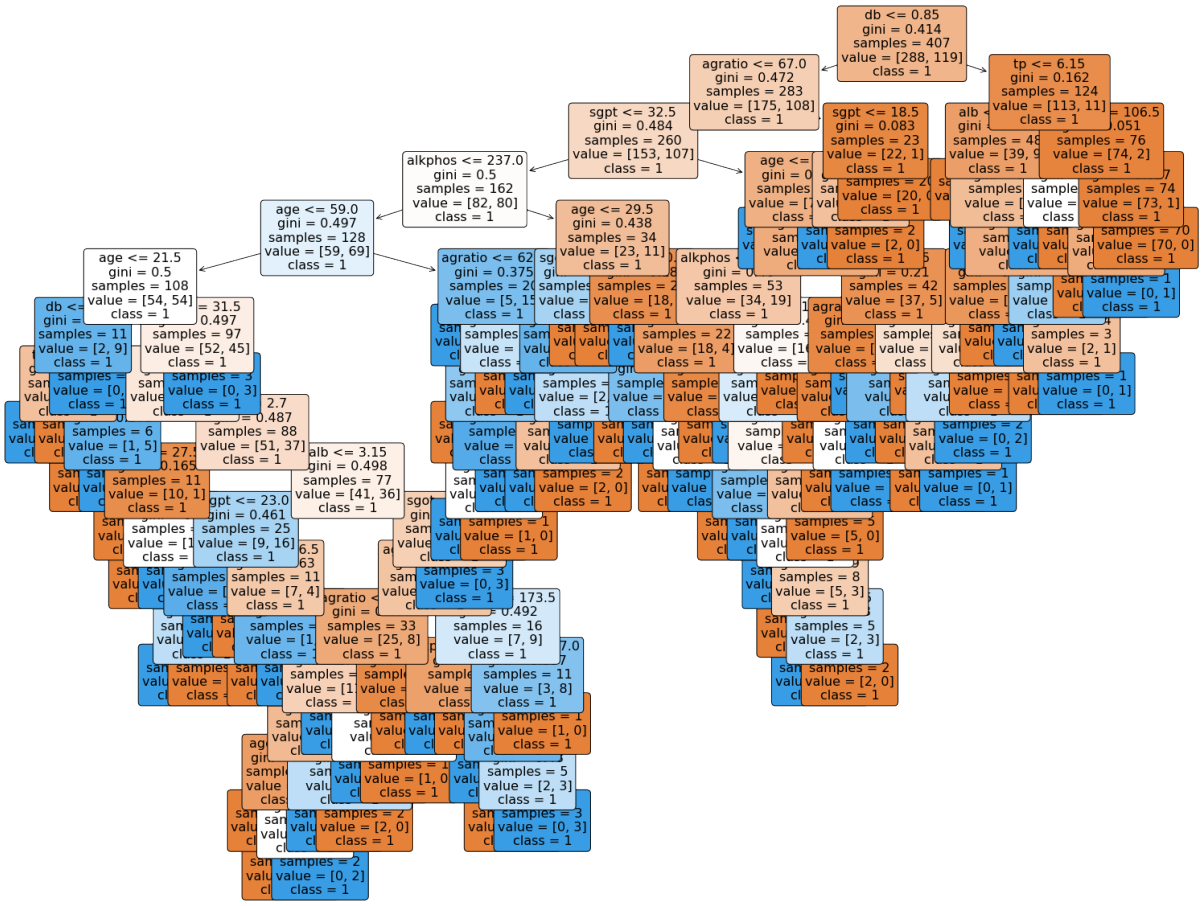
In [286]:
```
dt=DecisionTreeClassifier()
```

In [287]:
```
dt.fit(x_train,y_train)
```

Out[287]:
```
DecisionTreeClassifier()
```

In [288]:
```
pred=dt.predict(x_test)
```

In [292]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(25,20))
x.columns = x.columns.astype(str)
y=y.astype(str)
a = plot_tree(dt,feature_names=x.columns,class_names=y,filled=True,
```



Display the classification metrics. (3 Marks)

In [289]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

In [290]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           1       0.78      0.82      0.80       127
           2       0.45      0.40      0.42        48

    accuracy                           0.70       175
   macro avg       0.62      0.61      0.61       175
weighted avg       0.69      0.70      0.70       175
```

In [291]:
```
print(confusion_matrix(y_test,pred))
```

```
[[104  23]
 [ 29  19]]
```

Perform cross validation using cross (5 Marks)

In [293]:
```
from sklearn.model_selection import cross_val_score
```

In [294]:
```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=300,random_state=0)
rf.fit(x_train, y_train)
```

Out[294]:
```
RandomForestClassifier(n_estimators=300, random_state=0)
```

In [295]:
```
rfpred= rf.predict(x_test)
```

In [296]:
```
print(confusion_matrix(y_test,rfpred))
```

```
[[121   6]
 [ 39   9]]
```

In [297]:
```
print(classification_report(y_test,rfpred))
```

```
              precision    recall  f1-score   support

           1       0.76      0.95      0.84       127
           2       0.60      0.19      0.29        48

    accuracy                           0.74       175
   macro avg       0.68      0.57      0.56       175
weighted avg       0.71      0.74      0.69       175
```

In [298]:
```
all_accuracies=cross_val_score(estimator=rf,X=x_train,y=y_train,cv=
```

In [299]:
```
all_accuracies
```

Out[299]:
```
array([0.74390244, 0.7195122 , 0.7654321 , 0.66666667, 0.74074074]
)
```

In [300]:
```
all_accuracies.mean()
```

Out[300]:
```
0.7272508280638361
```

In [ ]:
```

```