Machine Learning Lab

Lab FAT

S Shyam Sundaram
19BCE1560
December 6, 2021

Dr Abdul Quadir MD
L31+L32

Question number: 2
Question:
The children.csv dataset contains the information of around 2300 children that attended the emergency services with fever and were tested for serious bacterial infection. The outcome of the children infected has 4 categories: Not Applicable(no infection) / UTI / Pneum / Bact:

a. Build a model using wcc, age, prevAB, pct, and crp to predict the outcome.
b. Compute the confusion matrix .and calculate the accuracy, recall, precision and plot the graphs. Write your observations in a separate cell.
c. Reduce the depth of the tree and infer the observations
d. How does the model classify a child with 1 year of age, WCC=29, PCT=5, CRP=200 and no prevAB?
e. Calculate probability for any given input (Note while calculating the probability you are not supposed to use library function)

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sb
import math
from sklearn import preprocessing
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus
import os
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
%matplotlib inline
```

Reading data

In [2]:
```
df=pd.read_csv("children.csv")
df.head()
```

Out[2]:

	Unnamed: 0	id	fever_hours	age	sex	wcc	prevAB	sbi	pct	crp
0	1	57906	24.0	0.79	M	3.8	No	UTI	0.090000	17.700000
1	2	58031	48.0	1.91	F	25.3	Yes	UTI	4.400000	150.400000
2	3	58148	24.0	0.07	F	20.0	No	UTI	0.548136	47.359279
3	4	58169	72.0	0.95	M	6.0	No	UTI	0.310000	4.900000
4	5	58517	1.0	0.11	F	15.6	No	UTI	0.936872	31.394860

Drop unnecessary columns.

In [3]:
```
df=df.drop(['Unnamed: 0','id','sex','fever_hours'],axis=1)
df.head()
```

Out[3]:

	age	wcc	prevAB	sbi	pct	crp
0	0.79	3.8	No	UTI	0.090000	17.700000
1	1.91	25.3	Yes	UTI	4.400000	150.400000
2	0.07	20.0	No	UTI	0.548136	47.359279
3	0.95	6.0	No	UTI	0.310000	4.900000
4	0.11	15.6	No	UTI	0.936872	31.394860

In [4]:
```
df.dtypes
```

Out[4]:
```
age       float64
wcc       float64
prevAB     object
sbi        object
pct       float64
crp       float64
dtype: object
```

Encode object type column.

In [5]:
```
classes = {
    'No': 0,
    'Yes' : 1
    }
df = df.replace({'prevAB': classes})
```

In [6]:
```
X = df.drop('sbi',axis=1)
Y = df['sbi']
```

In [7]:
```
X = np.array (X)
Y = np.array (Y)
```

Split train and test.

In [8]:
```
from sklearn.model_selection import train_test_split
train_x, test_x, train_y, test_y = train_test_split(X,Y, test_size=0.30, random_state=42)
```

a. Build Model

In [9]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [10]:
```
tree=DecisionTreeClassifier()
```

In [11]:
```
tree = DecisionTreeClassifier(criterion='entropy')
tree.fit(train_x, train_y)
```

Out[11]:
```
DecisionTreeClassifier(criterion='entropy')
```

A model is built to predict SBI based on age, WCC, PCT, CRP and prevAB. We evaluate it below.

b. Compute confusion matrix, accuracy, recall and precision

In [12]:
```
df['sbi'].unique()
```

Out[12]:
```
array(['UTI', 'Pneu', 'Bact', 'NotApplicable'], dtype=object)
```

In [13]:
```
predict = tree.predict(test_x)
```

In [14]:
```
acc=metrics.accuracy_score(test_y, predict)
prec=precision_score(test_y,predict, average=None)
rec=recall_score(test_y,predict,average=None)
print('Accuracy: ',acc)
print('Precision: ',prec)
print('Recall: ',rec)
confusion_matrix(test_y, predict)
```

Out[14]:
```
Accuracy:  0.6156028368794326
Precision:  [0.         0.761979   0.14285714 0.28125   ]
Recall:  [0.         0.77259413 0.14117647 0.26732673]
array([[  0,   5,   1,   2],
       [  2, 395,  56,  58],
       [  3,  61,  12,   9],
       [  1,  58,  15,  27]], dtype=int64)
```

Observations:

We see that we get an accuracy of about 0.62. It seems to not predict any 'UTI' class correctly. Hence it has 0 precision and recall for UTI. It has the highest precision and recall for Pneu class. The remaining classes are also detected poorly.

c. Reduce the depth of the tree and infer the observations

In [15]:
```
tree = DecisionTreeClassifier(criterion='entropy', max_depth = 5)
tree.fit(train_x, train_y)
```

Out[15]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

In [16]:
```
predict = tree.predict(test_x)
```

In [17]:
```
acc=metrics.accuracy_score(test_y, predict)
prec=precision_score(test_y,predict, average=None)
rec=recall_score(test_y,predict,average=None)
print('Accuracy: ',acc)
print('Precision: ',prec)
print('Recall: ',rec)
confusion_matrix(test_y, predict)
```

Out[17]:
```
Accuracy:  0.7163120567375887
Precision:  [0.         0.74355083 0.14285714 0.4       ]
Recall:  [0.         0.95890411 0.01176471 0.13861386]
array([[  0,   7,   0,   1],
       [  2, 490,   5,  14],
       [  1,  77,   1,   6],
       [  1,  85,   1,  14]], dtype=int64)
```

Observations:

The accuracy is better than the model which was not limited in depth. However, there is still some drawback. UTI is not detected at all but that of Pneu is the highest and has grown.

d. How does the model classify a child with 1 year of age, WCC=29, PCT=5, CRP=200 and no prevAB?

In [18]:
```
# age   wcc     prevAB  sbi     pct     crp
test=np.array([1,29,0,5,200])
test=test.reshape(1,-1)
pred=tree.predict(test)
pred
```

Out[18]:
```
array(['NotApplicable'], dtype=object)
```

It is classified as 'Not Applicable'.

Inferences

The model is a decision tree. Initially, the tree was trained without any limit on depth. It yielded an accuracy of 0.61. Later, when a limit was defined, it was observed that a limit of 5, i.e. depth of 5, yielded the best accuracy of 0.71. But, there is a limit beyond which the accuracy drops again.