

Assignment : Decision Tree Postprunning

- Diabetes Dataset
-

Importing Libraries

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
```

```
In [ ]: data = pd.read_csv('./Dataset/diabetes.csv')
```

```
In [ ]: data.head()
```

```
Out[ ]:
```

	glucose	bloodpressure	diabetes
0	40	85	0
1	40	92	0
2	45	63	1
3	45	80	0
4	40	73	1

Splitting Data into train and test

```
In [ ]: X= data.iloc[:, :-1]
y = data['diabetes'].values
```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42)
```

```
In [ ]: X_train
```

Out[]:

	glucose	bloodpressure
903	45	83
544	50	80
465	50	70
529	45	77
468	50	88
...
106	45	75
270	55	67
860	35	78
435	40	93
102	45	95

666 rows × 2 columns

Postprunning

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

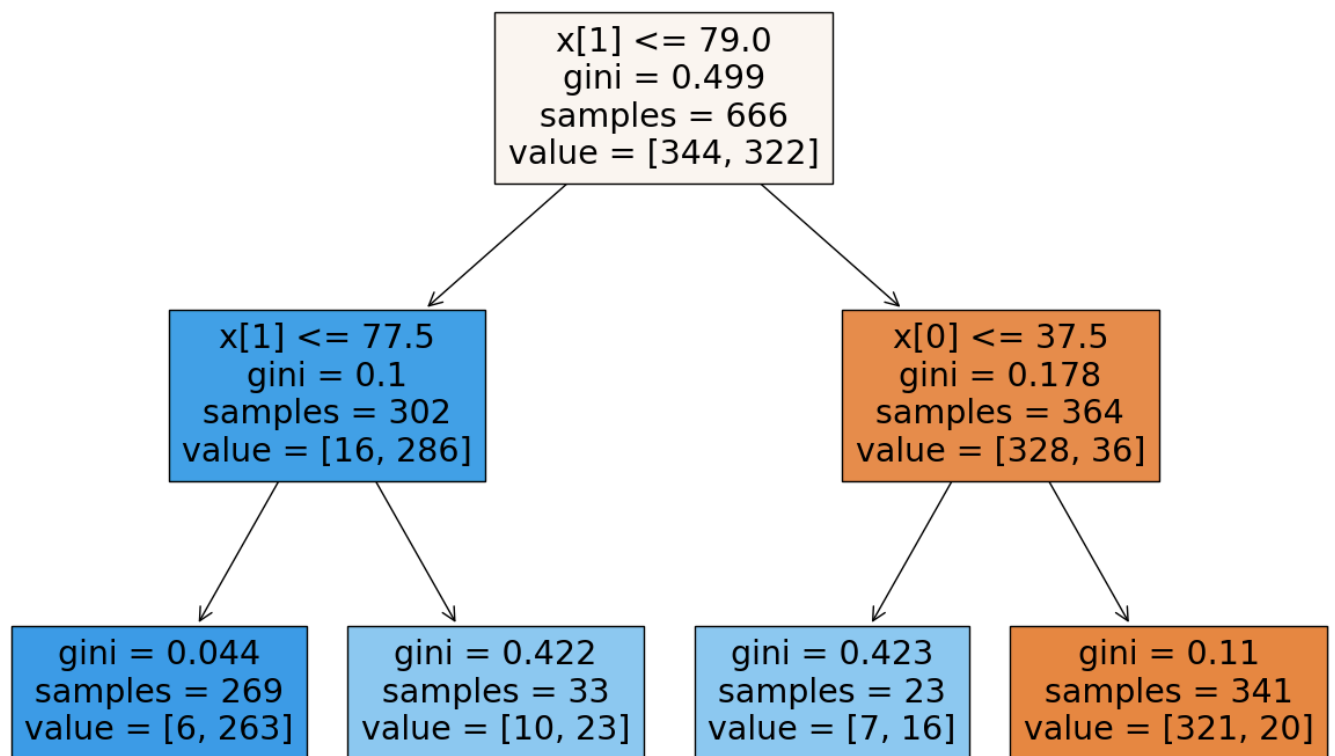
In []: `treemodel=DecisionTreeClassifier(max_depth=2)`

In []: `treemodel.fit(X_train,y_train)`

Out[]: `DecisionTreeClassifier`
`DecisionTreeClassifier(max_depth=2)`

In []: `from sklearn import tree`
`plt.figure(figsize=(15,10))`
`tree.plot_tree(treemodel,filled=True)`

Out[]: `[Text(0.5, 0.8333333333333334, 'x[1] <= 79.0\ngini = 0.499\nsamples = 666\nvalue = [344, 322]'),`
`Text(0.25, 0.5, 'x[1] <= 77.5\ngini = 0.1\nsamples = 302\nvalue = [16, 286]'),`
`Text(0.125, 0.16666666666666666, 'gini = 0.044\nsamples = 269\nvalue = [6, 263]'),`
`Text(0.375, 0.16666666666666666, 'gini = 0.422\nsamples = 33\nvalue = [10, 23]'),`
`Text(0.75, 0.5, 'x[0] <= 37.5\ngini = 0.178\nsamples = 364\nvalue = [328, 36]'),`
`Text(0.625, 0.16666666666666666, 'gini = 0.423\nsamples = 23\nvalue = [7, 16]'),`
`Text(0.875, 0.16666666666666666, 'gini = 0.11\nsamples = 341\nvalue = [321, 20]')]`



```
In [ ]: y_pred=treemodel.predict(X_test)
```

```
In [ ]: y_pred
```

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

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

```
In [ ]: score=accuracy_score(y_pred,y_test)
        print(score)
```

```
0.9179331306990881
```

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

	precision	recall	f1-score	support
0	0.92	0.91	0.91	154
1	0.92	0.93	0.92	175
accuracy			0.92	329
macro avg	0.92	0.92	0.92	329
weighted avg	0.92	0.92	0.92	329
