

LAB4 : Implement Decision tree algorithm for classification

Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

import Dataset

```
dataset = pd.read_csv('Result.csv')
```

EDA Steps

```
dataset.head()

   initial payment  last payment  credit score  house number  results
0              201          10018        250         3046     yes
1              205          10016        395         3044     yes
2              257          10129        109         3251     yes
3              246          10064        324         3137     yes
4              117          10115        496         3094     yes

dataset.shape
(1000, 5)

dataset.columns
Index(['initial payment', 'last payment', 'credit score', 'house
number',
       'results'],
      dtype='object')

dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   initial payment  1000 non-null   int64  
 1   last payment     1000 non-null   int64  
 2   credit score     1000 non-null   int64  
 3   house number     1000 non-null   int64  
 4   results          1000 non-null   object 

```

```

dtypes: int64(4), object(1)
memory usage: 39.2+ KB

dataset.describe()

   initial payment  last payment  credit score  house number
count      1000.00000    1000.00000  1000.000000  1000.000000
mean       294.34300    12465.88400   528.042000  4027.011000
std        115.81539    1440.15617   259.796059   565.164179
min        100.00000    10005.00000   100.000000  3003.000000
25%       195.00000    11201.50000   302.000000  3545.000000
50%       289.50000    12450.00000   516.500000  4041.500000
75%       398.00000    13678.25000   753.500000  4507.000000
max       500.00000    14996.00000   997.000000  5000.000000

```

Preprocessing Steps

Step 1 : Separate input and output variables

```

X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

```

Step 4 : splitting Data into training and testing

```

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 = 0)

print(X_train.shape)
print(X_test.shape)

(750, 4)
(250, 4)

# Step 5 : Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)

print(X_test)

[[ 2.10451644e-01  1.75517538e+00 -9.79776424e-01  1.43497147e+00]
 [-1.16903831e+00  1.09443174e+00 -1.49655229e+00  1.21452245e+00]
 [ 1.30568307e+00 -5.42759691e-01  1.58879104e+00 -1.50493670e+00]
 [ 1.27224088e+00 -3.82812235e-01  1.40887648e+00  3.71525384e-01]
 [-1.66231048e+00  6.23669357e-01 -1.14437911e+00  6.78390424e-01]
 [ 1.40600961e+00  1.21526541e+00  1.16676507e-02  1.30622924e+00]
 [-5.08555244e-01 -1.26706324e+00 -1.21711053e+00 -1.56313524e+00]

```

| | | | |
|-------------------|-----------------|-----------------|------------------|
| [-1.15231722e+00 | 4.32486746e-02 | -1.50038026e+00 | -1.79240223e+00] |
| [-5.33636879e-01 | -3.63953801e-01 | -3.64422687e-03 | -7.80100312e-01] |
| [-4.91834153e-01 | 1.27812685e+00 | -1.18648677e+00 | -9.17660502e-01] |
| [-1.65394994e+00 | 3.63143587e-01 | -1.70326264e+00 | 9.64050031e-02] |
| [2.68975460e-01 | -4.89481124e-02 | -1.60756341e+00 | 1.23921274e+00] |
| [4.32407398e-02 | 1.19291467e+00 | 7.73433561e-01 | -1.01289448e+00] |
| [9.79621803e-01 | -1.46612448e+00 | 1.37825273e+00 | 3.57416646e-01] |
| [-7.25929419e-01 | 6.59989303e-01 | -2.52462238e-01 | -8.33008077e-01] |
| [-1.06871177e+00 | -1.58556123e+00 | 1.04139142e+00 | -1.01994885e+00] |
| [-7.51011055e-01 | 1.52608033e+00 | -3.44333503e-01 | -2.88100419e-02] |
| [-1.49509958e+00 | -1.50104751e+00 | 1.07584314e+00 | 4.35014702e-01] |
| [-1.64558939e+00 | -1.41653379e+00 | -8.72593281e-01 | -1.48730078e+00] |
| [-7.42650509e-01 | -1.26846016e+00 | -1.49107064e-01 | -7.99499826e-01] |
| [-6.50684512e-01 | -1.63165962e+00 | -1.18265880e+00 | -5.91395948e-01] |
| [-8.68058688e-01 | 3.82700481e-01 | -7.00334658e-01 | -7.43064876e-01] |
| [1.43945179e+00 | -1.28173091e+00 | 6.62422449e-01 | 8.28295759e-01] |
| [1.47289397e+00 | -3.99575287e-01 | -6.54399025e-01 | -7.90681865e-01] |
| [1.16355380e+00 | 1.11887786e+00 | 1.44332821e+00 | -4.87344010e-01] |
| [1.35584634e+00 | 6.51607777e-01 | -7.88377954e-01 | 8.38877312e-01] |
| [-6.25602877e-01 | -1.50174597e+00 | 1.56582323e+00 | -1.76947553e+00] |
| [7.66829206e-02 | -1.11549731e+00 | -8.87905158e-01 | -6.53121675e-01] |
| [1.41437015e+00 | 6.97706170e-01 | 1.35911288e+00 | -4.90871194e-01] |
| [-8.34616507e-01 | -1.72036411e+00 | -6.46743086e-01 | -1.82943766e+00] |
| [-9.88885287e-02 | 6.35543185e-01 | 1.91582213e-01 | 1.03639964e+00] |
| [4.52907455e-01 | 1.20339158e+00 | -5.24248065e-01 | -1.20336244e+00] |
| [-6.54463479e-02 | -3.42301526e-01 | 1.58879104e+00 | 1.00994575e+00] |
| [-1.39477304e+00 | 6.81641579e-01 | 1.11795081e+00 | 4.75577322e-01] |
| [7.45526537e-01 | -1.62797175e-01 | 4.05948499e-01 | -9.17660502e-01] |
| [1.44781233e+00 | 8.08761391e-01 | -7.42442321e-01 | -1.13462467e-01] |
| [-1.71247375e+00 | 1.55122491e+00 | -2.33322391e-01 | 9.58801581e-01] |
| [5.03070726e-01 | -9.01768392e-01 | -2.75430054e-01 | 9.25293329e-01] |
| [5.99618302e-02 | 9.49151953e-01 | 2.22205968e-01 | 1.13692439e+00] |
| [-1.25264377e+00 | -4.06559892e-01 | 1.16388644e+00 | -1.41322991e+00] |
| [-9.93466866e-01 | -3.13664645e-01 | 2.41345815e-01 | -1.73420368e+00] |
| [4.52907455e-01 | 1.35775135e+00 | 1.38973663e+00 | 1.33444672e+00] |
| [-7.76092690e-01 | 5.83158647e-01 | 2.33689876e-01 | -3.72710518e-01] |
| [1.34748579e+00 | 1.89226921e-01 | 1.65386652e+00 | 7.17147125e-02] |
| [1.27224088e+00 | -1.46682294e+00 | -1.10227144e+00 | 1.56900448e+00] |
| [4.32407398e-02 | 4.05749678e-01 | 9.30380307e-01 | 8.81203525e-01] |
| [-1.62886830e+00 | 1.61268944e+00 | 1.39356460e+00 | -1.69364106e+00] |
| [-2.99541614e-01 | -1.04425434e+00 | 3.71496774e-01 | -9.45877977e-01] |
| [-2.74459978e-01 | -3.00393895e-01 | 7.61949653e-01 | -1.26685175e+00] |
| [-9.93466866e-01 | -2.80138540e-01 | 1.58879104e+00 | 8.07132653e-01] |
| [-2.07575616e-01 | -8.18651592e-01 | 1.18302629e+00 | -7.28998465e-02] |
| [-1.23970164e-01 | -7.76045501e-01 | -1.59225153e+00 | 8.33586536e-01] |
| [3.69302003e-01 | -9.74408285e-01 | -6.62054964e-01 | 4.47359848e-01] |
| [1.48125452e+00 | -8.59665193e-02 | 5.70551183e-01 | -1.10636487e+00] |
| [-2.24296707e-01 | 1.63434171e+00 | -1.89561045e-02 | -1.01642166e+00] |
| [5.99618302e-02 | 5.86148058e-02 | -6.23775270e-01 | 7.36588966e-01] |

| | | | |
|-------------------|-----------------|-----------------|------------------|
| [3.52580912e-01 | -1.22515561e+00 | -1.46592854e+00 | 1.60251273e+00] |
| [2.52254370e-01 | -1.04844510e+00 | -1.72623046e+00 | 1.24803070e+00] |
| [5.53233997e-01 | 1.54284338e+00 | 9.68660001e-01 | 6.16664697e-01] |
| [-9.68385230e-01 | -4.64532115e-01 | 8.61476858e-01 | -7.32483323e-01] |
| [7.12084356e-01 | 4.09940441e-01 | 8.57648888e-01 | -1.07991098e+00] |
| [1.47289397e+00 | 2.68851419e-01 | -5.05108218e-01 | 3.66234607e-01] |
| [-7.34289964e-01 | -1.52828747e+00 | -5.16592126e-01 | 7.89496731e-01] |
| [1.43567282e-01 | -7.32740950e-01 | 4.40400224e-01 | -1.30212360e+00] |
| [7.70608173e-01 | -3.30427697e-01 | 1.58496307e+00 | -9.88204190e-01] |
| [-1.01854850e+00 | -1.10921116e+00 | 9.99283756e-01 | -4.23854691e-01] |
| [-1.51182067e+00 | -1.18534336e+00 | 8.00229347e-01 | 7.24243820e-01] |
| [-6.67405603e-01 | 5.44240428e-02 | 1.24427380e+00 | -4.13273138e-01] |
| [-1.65394994e+00 | -1.10222656e+00 | 9.99283756e-01 | 3.00981696e-01] |
| [7.78968718e-01 | 8.04570628e-01 | -1.60756341e+00 | -9.30005648e-01] |
| [1.07158780e+00 | -9.29511244e-02 | 1.16388644e+00 | 1.37148215e+00] |
| [-4.41670882e-01 | -1.13645112e+00 | 1.83742531e-04 | 1.09636177e+00] |
| [-2.32657252e-01 | 3.25426720e-01 | -1.60590972e-01 | 1.27977536e+00] |
| [-8.68058688e-01 | -1.06031893e+00 | 5.76032835e-02 | 4.05033635e-01] |
| [-1.04363014e+00 | 1.44366199e+00 | -8.49625464e-01 | 7.98314692e-01] |
| [1.44781233e+00 | -1.21328178e+00 | 7.84917470e-01 | 1.44555302e+00] |
| [4.32407398e-02 | -1.91434056e-01 | 1.30934928e+00 | -1.24745224e+00] |
| [8.37492534e-01 | 4.34386559e-01 | 7.88745439e-01 | 8.70621972e-01] |
| [1.45617288e+00 | -6.34956478e-01 | 1.31700522e+00 | 1.39617244e+00] |
| [1.36420688e+00 | -4.96465729e-02 | 6.58594479e-01 | 7.63042848e-01] |
| [-1.31116758e+00 | 1.48017751e-01 | -1.22476647e+00 | -1.78005708e+00] |
| [1.57322051e+00 | -1.18743874e+00 | 1.50457572e+00 | 1.40851759e+00] |
| [1.48125452e+00 | -8.45193091e-01 | -1.07930363e+00 | 1.17043264e+00] |
| [-1.70411321e+00 | 1.62875403e+00 | 1.19833816e+00 | 5.79629261e-01] |
| [7.20444901e-01 | 1.03995182e+00 | 3.48528958e-01 | 6.23719066e-01] |
| [-1.31116758e+00 | 6.80244658e-01 | 1.53519947e+00 | -5.79050803e-01] |
| [-4.83473608e-01 | -1.39697689e+00 | 1.23661786e+00 | 5.17903535e-01] |
| [-8.76419233e-01 | -2.80138540e-01 | -1.60590972e-01 | 8.84730709e-01] |
| [9.71261258e-01 | -4.70818259e-01 | 1.50457572e+00 | 1.77530244e-01] |
| [-3.24623249e-01 | 1.03017337e+00 | -1.22093850e+00 | 1.18101420e+00] |
| [5.53233997e-01 | -2.99695435e-01 | 1.45098415e+00 | -6.33722161e-01] |
| [1.16355380e+00 | 7.52186090e-01 | -1.34343352e+00 | -1.66542359e+00] |
| [1.68648918e-01 | 1.20269312e+00 | -1.09078754e+00 | 1.12810643e+00] |
| [-1.47837849e+00 | -9.46469865e-01 | 2.29861907e-01 | 5.76102077e-01] |
| [-8.51337597e-01 | 7.08881539e-01 | 8.95928582e-01 | -9.45877977e-01] |
| [8.62574170e-01 | -2.26357081e-01 | -7.38614352e-01 | 5.12612758e-01] |
| [-7.09208329e-01 | 5.09317402e-02 | -7.92205923e-01 | 8.82967117e-01] |
| [1.60666269e+00 | -1.47310909e+00 | -4.63000555e-01 | -2.51022657e-01] |
| [-1.53690230e+00 | -2.85726225e-01 | 9.34208276e-01 | -7.55410021e-01] |
| [1.17191434e+00 | 1.64780803e-01 | -3.97925075e-01 | 6.55463725e-01] |
| [1.56485997e+00 | 9.80582676e-01 | -1.10227144e+00 | 7.52418969e-02] |
| [-7.51011055e-01 | 1.69301239e+00 | 3.63840836e-01 | -1.14340030e+00] |
| [1.23879870e+00 | -1.22585407e+00 | 9.20550081e-02 | -1.71480417e+00] |
| [-7.76092690e-01 | 1.33260677e+00 | -3.86441167e-01 | 4.40305479e-01] |
| [5.99618302e-02 | -1.14603400e-01 | 7.29151611e-02 | -1.48553719e+00] |

| | | | |
|-------------------|-----------------|-----------------|------------------|
| [2.43893825e-01 | 6.02715541e-01 | -1.75902850e-01 | -4.48544982e-01] |
| [-7.76092690e-01 | 7.32629196e-01 | 2.71969570e-01 | -4.73235272e-01] |
| [1.58158106e+00 | 9.30991980e-01 | 5.20787581e-01 | -5.12034300e-01] |
| [1.06322725e+00 | 1.16497625e+00 | -3.90269136e-01 | 1.42791710e+00] |
| [7.62247627e-01 | -9.68122141e-01 | -1.12141129e+00 | 2.93885001e-02] |
| [-9.34943049e-01 | -1.44307529e+00 | -1.25539022e+00 | 4.93213244e-01] |
| [7.37165992e-01 | -3.41603065e-01 | 1.48926384e+00 | 6.30773435e-01] |
| [-1.06035123e+00 | 5.24487965e-01 | -1.49272432e+00 | -8.77097882e-01] |
| [-1.52854176e+00 | -1.62797175e-01 | -1.04102393e+00 | 1.30270206e+00] |
| [-4.75113063e-01 | -1.43888452e+00 | -5.34078291e-02 | 1.43320788e+00] |
| [1.36420688e+00 | 1.18179518e-02 | -1.72074880e-01 | -1.39030321e+00] |
| [-2.36436218e-02 | -5.19710495e-01 | 1.11194855e-01 | 1.52667826e+00] |
| [9.54540167e-01 | -1.16508800e+00 | 1.12177878e+00 | 1.03992682e+00] |
| [1.130111162e+00 | 4.29497335e-01 | -1.72623046e+00 | -6.93684295e-01] |
| [-2.91181068e-01 | 8.00379865e-01 | -1.59607950e+00 | -1.21394399e+00] |
| [-1.61214721e+00 | 1.51490496e+00 | -9.22356883e-01 | -7.36010507e-01] |
| [1.05486671e+00 | 1.06719178e+00 | 2.26033937e-01 | 4.57941401e-01] |
| [9.21097986e-01 | -1.70080721e+00 | -1.74154233e+00 | 1.23921274e+00] |
| [-1.08543286e+00 | 1.03576106e+00 | 6.16486816e-01 | 1.52315108e+00] |
| [-1.32330710e-01 | -1.41094610e+00 | 3.82980683e-01 | -1.22981632e+00] |
| [-7.00847783e-01 | 9.18419691e-01 | 1.16005847e+00 | 2.14565679e-01] |
| [-1.23970164e-01 | -5.72095033e-01 | 4.67196009e-01 | -2.88100419e-02] |
| [2.18812189e-01 | -5.21107416e-01 | -6.46743086e-01 | -4.15036730e-01] |
| [8.20771444e-01 | 1.43039124e+00 | -6.23775270e-01 | 5.70811300e-01] |
| [6.03397269e-01 | -5.76285796e-01 | 1.33231709e+00 | 1.49140642e+00] |
| [8.96016351e-01 | 1.52208514e-01 | 8.92100613e-01 | -1.11518283e+00] |
| [-7.84453235e-01 | 1.20079331e-01 | -1.21711053e+00 | 1.43144429e+00] |
| [5.99618302e-02 | 2.96091378e-01 | 1.02607954e+00 | -5.68469250e-01] |
| [2.77336006e-01 | 5.92937094e-01 | 1.60958457e-01 | 7.34783047e-02] |
| [-7.92813781e-01 | 1.45972658e+00 | 6.54766510e-01 | -1.72362213e+00] |
| [9.62900712e-01 | 1.04693642e+00 | -6.42915117e-01 | -1.61427941e+00] |
| [1.47289397e+00 | 2.64856225e-02 | -6.16119331e-01 | -1.87705465e+00] |
| [-1.41149413e+00 | -1.46891832e+00 | -1.27835804e+00 | -1.51375466e+00] |
| [9.79621803e-01 | 8.72516866e-02 | 9.58829775e-02 | -3.25093529e-01] |
| [-6.54463479e-02 | 1.17894546e+00 | 6.08830877e-01 | -7.99542152e-02] |
| [6.11757814e-01 | -9.70915983e-01 | -5.08936188e-01 | -1.31094156e+00] |
| [-1.69575266e+00 | 4.52546532e-01 | -5.51043851e-01 | 3.34489948e-01] |
| [-1.21084104e+00 | -2.31050736e-02 | 1.09498299e+00 | -1.18753243e-01] |
| [-1.67903157e+00 | -2.47310897e-01 | 5.20787581e-01 | -1.66366000e+00] |
| [1.51927827e-01 | -1.01561746e+00 | 4.97819765e-01 | 2.83345775e-01] |
| [-2.41017797e-01 | -1.32064913e-01 | -3.29021626e-01 | -2.79240132e-01] |
| [-1.36133085e+00 | -9.27611431e-01 | 4.32744285e-01 | 1.27624817e+00] |
| [-4.83473608e-01 | -3.34618460e-01 | -9.03217036e-01 | 2.88636551e-01] |
| [-1.16903831e+00 | -1.70918874e+00 | -1.31280976e+00 | 3.83870529e-01] |
| [-1.21084104e+00 | -1.56181357e+00 | -9.83604393e-01 | -6.61939636e-01] |
| [-1.60378667e+00 | -1.04844510e+00 | -5.39559943e-01 | 1.13868798e+00] |
| [5.11431271e-01 | -1.71337950e+00 | -1.62670325e+00 | -4.48544982e-01] |
| [1.59830215e+00 | 1.28460857e-01 | -1.73388640e+00 | -8.70043513e-01] |
| [2.68975460e-01 | 1.01061648e+00 | -1.29749788e+00 | 1.68712283e-01] |

| | | | |
|------------------|-----------------|-----------------|------------------|
| [2.77336006e-01 | 1.13354553e+00 | 8.76788735e-01 | 1.42968069e+00] |
| [-8.34616507e-01 | 1.70348930e+00 | 1.06053127e+00 | 6.36064211e-01] |
| [-1.99215071e-01 | -8.87603613e-02 | -1.63053122e+00 | 4.56177808e-01] |
| [1.07158780e+00 | -4.63135194e-01 | 2.37517845e-01 | -5.77287211e-01] |
| [7.70608173e-01 | 5.16804899e-01 | 1.68614396e-01 | 1.50022438e+00] |
| [4.86349636e-01 | 7.28438433e-01 | -1.74919827e+00 | -1.40441194e+00] |
| [3.48801946e-02 | 1.18179518e-02 | 9.87799848e-01 | 9.81685953e-02] |
| [6.70281630e-01 | 1.72514158e+00 | -1.14437911e+00 | 1.04169041e+00] |
| [1.04650616e+00 | 3.77811258e-01 | -8.19001709e-01 | 7.27771005e-01] |
| [1.41437015e+00 | 1.03506260e+00 | 2.83453478e-01 | 1.52797627e-02] |
| [1.03814562e+00 | 1.47928348e+00 | 1.63472668e+00 | -8.66516329e-01] |
| [3.35859822e-01 | 7.39809370e-02 | -5.72357985e-02 | -1.46437408e+00] |
| [5.86676178e-01 | -9.29706813e-01 | 1.63855465e+00 | -2.35150327e-01] |
| [3.27499277e-01 | 1.49255423e+00 | 4.28916315e-01 | 5.40787907e-02] |
| [-1.29444649e+00 | -7.36931713e-01 | -1.28601398e+00 | 1.21981322e+00] |
| [-5.25276334e-01 | -7.52297844e-01 | 1.02990751e+00 | 6.36064211e-01] |
| [9.54540167e-01 | 1.67764626e+00 | 1.31700522e+00 | -8.64752737e-01] |
| [-7.51011055e-01 | 1.03157029e+00 | 5.39927428e-01 | -2.21041590e-01] |
| [-1.16067777e+00 | -8.34716184e-01 | -1.27453007e+00 | -4.94398378e-01] |
| [-1.62886830e+00 | 1.41432665e+00 | 5.43755397e-01 | -7.51882837e-01] |
| [-1.49051800e-01 | 2.68152958e-01 | 4.36572254e-01 | -9.97022150e-01] |
| [-7.59371600e-01 | 5.41949477e-01 | 8.88272643e-01 | 4.52650624e-01] |
| [1.04650616e+00 | -1.17556491e+00 | 1.37059679e+00 | -3.48020227e-01] |
| [-6.33963422e-01 | 1.08325637e+00 | 4.99473447e-02 | -1.44321097e+00] |
| [2.02091099e-01 | 6.76752355e-01 | 7.88745439e-01 | 1.17748701e+00] |
| [-4.16589246e-01 | -1.40954918e+00 | -6.65882933e-01 | 5.16139943e-01] |
| [-1.51182067e+00 | 1.42200972e+00 | -1.28601398e+00 | -4.22091099e-01] |
| [-1.14395668e+00 | 1.61967404e+00 | 1.03373548e+00 | 1.50904234e+00] |
| [-9.09861414e-01 | -1.54415649e-01 | -8.19001709e-01 | 4.94976837e-01] |
| [1.61502324e+00 | -7.39027094e-01 | -6.00807453e-01 | 6.11331594e-02] |
| [1.68648918e-01 | 1.70139392e+00 | -1.35108946e+00 | 1.52138749e+00] |
| [2.02091099e-01 | 1.07976407e+00 | -4.17064922e-01 | 1.10870692e+00] |
| [1.17191434e+00 | 1.72463869e-01 | 1.00693969e+00 | 5.81392853e-01] |
| [1.23043816e+00 | -3.33920000e-01 | -1.23242240e+00 | 6.04319552e-01] |
| [-4.91834153e-01 | 4.46260388e-01 | 6.43282602e-01 | 9.72910318e-01] |
| [-1.26100431e+00 | -1.51292134e+00 | 7.84917470e-01 | 8.14187022e-01] |
| [6.78642175e-01 | 5.47537161e-01 | 1.02225157e+00 | 7.78915178e-01] |
| [1.46453342e+00 | 7.22850749e-01 | -7.69238107e-01 | 9.87019056e-01] |
| [1.31404361e+00 | -1.13505420e+00 | 5.39927428e-01 | 1.59722195e+00] |
| [8.45853079e-01 | -1.60092736e+00 | -1.52334808e+00 | -1.78711145e+00] |
| [8.12410899e-01 | 1.20758234e+00 | 1.06435924e+00 | -1.44497457e+00] |
| [8.96016351e-01 | 4.69309584e-01 | -1.47358448e+00 | 7.45406927e-01] |
| [1.54813888e+00 | 1.03506260e+00 | 2.79625509e-01 | -1.29154205e+00] |
| [1.37256743e+00 | -4.91073614e-01 | -1.14437911e+00 | -1.28272408e+00] |
| [6.95363266e-01 | 7.61266077e-01 | 7.42809806e-01 | -1.18925370e+00] |
| [-4.41670882e-01 | -8.43796170e-01 | -1.61904731e+00 | -4.18563914e-01] |
| [1.02978507e+00 | 1.95513065e-01 | 1.18850794e-01 | 7.50697703e-01] |
| [5.19791816e-01 | -9.78403479e-02 | 1.07366886e-01 | 6.46180176e-03] |
| [-3.58065430e-01 | -1.25030019e+00 | 2.06894090e-01 | 1.50198797e+00] |

```

[-1.64558939e+00  1.28231762e+00  1.45098415e+00  1.24273992e+00]
[-6.92487238e-01  3.89685087e-01  1.49309181e+00  1.10870692e+00]
[ 5.03070726e-01  5.32171030e-01  7.04530112e-01  -8.70043513e-01]
[-1.52854176e+00  1.30536681e+00  -8.68765311e-01  5.40787907e-02]
[-1.33624922e+00  -4.58944430e-01  1.87754243e-01  1.37324574e+00]
[ 8.62574170e-01  -4.04464511e-01  -1.24773428e+00  1.45966176e+00]
[ 1.02978507e+00  1.73561849e+00  3.44700989e-01  -6.84866334e-01]
[-1.41985467e+00  -1.20490025e+00  -9.93434619e-02  -4.25618283e-01]
[ 1.15519325e+00  6.84435421e-01  9.72487970e-01  -1.58077116e+00]
[-1.00182741e+00  -1.45634604e+00  -1.61521934e+00  -6.16086239e-01]
[ 6.45199995e-01  7.54979932e-01  1.32848912e+00  7.15425859e-01]
[-2.24296707e-01  1.56170182e+00  -1.07930363e+00  -2.70464497e-02]
[-1.64558939e+00  1.65808937e+00  2.64313631e-01  1.38382730e+00]
[ 1.58158106e+00  -1.52339824e+00  1.11795081e+00  -3.03930422e-01]
[ 1.43567282e-01  -1.54574898e+00  -1.41616494e+00  1.17043264e+00]
[-1.34460976e+00  -1.36205387e+00  -1.72240249e+00  -6.72521189e-01]
[ 8.96016351e-01  3.44285154e-01  -7.04162627e-01  -1.18753243e-01]
[-1.02690905e+00  1.66367705e+00  8.46164980e-01  -1.78711145e+00]
[ 1.26388034e+00  4.64420361e-01  -6.10637679e-02  -7.73045943e-01]
[-1.10215395e+00  1.59941869e+00  -3.74957259e-01  8.60040418e-01]
[-1.00182741e+00  -9.82789811e-01  -2.37150360e-01  6.25482658e-01]
[-8.01174326e-01  4.46958848e-01  -1.48889635e+00  -1.27390612e+00]
[-1.47001794e+00  -2.66169330e-01  4.51884132e-01  -1.11871001e+00]
[-1.33624922e+00  5.10518754e-01  -9.75948455e-01  1.58130730e-01]
[ 1.52305724e+00  -7.92808553e-01  3.21733172e-01  1.61838506e+00]
[-1.54526285e+00  -1.59324430e+00  -1.26687413e+00  -7.34246915e-01]
[-7.59371600e-01  -2.87821606e-01  1.53902744e+00  -6.28431384e-01]
[-6.75766148e-01  -3.68144564e-01  -4.89796341e-01  1.43497147e+00]
[ 8.87655805e-01  -8.57066920e-01  3.02593325e-01  2.93885001e-02]
[ 4.52907455e-01  -2.96203132e-01  1.15022825e-01  1.52138749e+00]
[ 5.36512907e-01  5.04931070e-01  4.44228193e-01  8.75912748e-01]
[ 5.11431271e-01  -1.41862917e+00  8.57648888e-01  -1.47848282e+00]
[-1.82493981e-01  -1.14622957e+00  -5.01280249e-01  -1.06051147e+00]
[ 9.34040110e-02  5.18900280e-01  -2.18010513e-01  2.16329272e-01]
[ 3.02417641e-01  -1.69102877e+00  1.23661786e+00  6.39591396e-01]
[ 9.29458532e-01  5.93635555e-01  -1.12141129e+00  8.84730709e-01]
[-1.40691255e-01  -1.42701069e+00  1.33997303e+00  -6.40818856e-02]
[-1.09379341e+00  1.44525448e-01  -5.81667606e-01  3.38017132e-01]
[ 9.96342893e-01  1.45223909e-01  -1.62287528e+00  -8.17178074e-02]
[-1.41149413e+00  -1.27195246e+00  1.22130598e+00  9.98898613e-03]
[-4.66752518e-01  -9.14340682e-01  1.08349908e+00  5.02031205e-01]
[ 1.40600961e+00  1.56100336e+00  -1.91214727e-01  1.14750594e+00]
[ 4.86349636e-01  -1.48827965e-01  1.22896192e+00  -7.83627496e-01]
[-1.32330710e-01  -2.94107751e-01  7.65777623e-01  -1.04816632e+00]
[-3.99868156e-01  -1.64143807e+00  -1.74154233e+00  -1.46437408e+00]
[ 1.61502324e+00  3.99463534e-01  -1.72074880e-01  -1.57900757e+00]
[ 1.25551979e+00  1.26136380e+00  -1.57311168e+00  -1.14869108e+00]

```

```
dataset['results'].value_counts()
```

```
results
No      521
yes     479
Name: count, dtype: int64
```

Train the Decision Tree Classification model on the Training set

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion =
'entropy', random_state= 80,
                                     splitter='random', ccp_alpha
=0.0001)

classifier.fit(X_train,y_train)

DecisionTreeClassifier(ccp_alpha=0.0001, criterion='entropy',
random_state=80,
                     splitter='random')

print(classifier.classes_)

['No' 'yes']

classifier.max_features_

4

classifier.tree_

<sklearn.tree._tree.Tree at 0x1e8ac5d7bb0>

classifier.n_outputs_

1
```

Predicting the Test set results

```
ypred = classifier.predict(X_test)

print(ypred)

['No' 'No' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'yes' 'yes' 'No' 'No' 'No' 'No'
 'yes' 'No' 'yes' 'No' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'No'
 'No'
 'yes' 'yes' 'No' 'yes' 'No' 'No' 'No' 'No' 'yes' 'No' 'No' 'yes' 'No'
 'yes' 'yes' 'No' 'No' 'yes' 'yes' 'No' 'No' 'Yes' 'yes' 'No' 'yes'
 'yes'
 'yes' 'yes' 'No' 'No' 'yes' 'yes' 'No' 'yes' 'No' 'No' 'yes' 'yes'
 'yes'
 'yes' 'yes' 'No' 'yes' 'No' 'No' 'yes' 'No' 'Yes' 'No' 'yes' 'yes'
 'No'
 'No' 'No' 'yes' 'yes' 'No' 'No' 'No' 'No' 'yes' 'yes' 'yes' 'yes' 'No'
```

```

'yes'
'No' 'No' 'yes' 'No' 'yes' 'No' 'yes' 'yes' 'No' 'No' 'No' 'yes' 'No'
'yes' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'No' 'No' 'No' 'No' 'yes' 'yes'
'yes'
'yes' 'No' 'No' 'No' 'No' 'yes' 'No' 'yes' 'No' 'yes' 'yes' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'yes' 'Yes' 'No' 'No' 'yes' 'No' 'yes'
'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'No'
'No' 'yes' 'yes' 'No' 'No' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'No'
'yes' 'yes' 'No' 'No' 'yes' 'No' 'No' 'No' 'yes' 'No' 'No' 'yes' 'No'
'No' 'yes' 'yes' 'No' 'No' 'yes' 'No' 'yes' 'No' 'yes' 'No' 'yes'
'yes'
'No' 'No' 'No' 'yes' 'No' 'yes' 'No' 'yes' 'No' 'No' 'No' 'No'
'No'
'yes' 'No' 'yes' 'No' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'yes' 'No'
'No'
'yes' 'No' 'yes' 'No' 'yes' 'No' 'No' 'No' 'yes' 'yes' 'yes' 'No'
'No'
'yes' 'yes' 'No' 'yes' 'No' 'yes' 'No' 'yes' 'yes' 'yes' 'No' 'yes'
'yes'
'yes' 'No' ]

```

Compare the predicted and Actual Output

```

print(np.concatenate((ypred.reshape(len(ypred),1),
                     y_test.reshape(len(y_test),1)),
                     1))

```

```

[[['No' 'No'],
 ['No' 'No'],
 ['yes' 'yes'],
 ['No' 'No'],
 ['No' 'No'],
 ['No' 'No'],
 ['yes' 'yes'],
 ['yes' 'yes'],
 ['yes' 'yes'],
 ['No' 'No'],
 ['No' 'No'],
 ['No' 'No'],
 ['No' 'No'],
 ['No' 'No'],
 ['No' 'No'],
 ['yes' 'yes'],
 ['No' 'No'],
 ['yes' 'yes'],
 ['No' 'No'],
 ['yes' 'yes'],
 ['yes' 'yes'],
 ['yes' 'yes'],
 ['yes' 'No'],
 ['yes' 'yes']]

```

```
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'yes']
['yes' 'yes']
['No' 'No']
['yes' 'yes']
['No' 'No']
['No' 'No']
['No' 'No']
['No' 'No']
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'yes']
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'No']
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'yes']
['yes' 'yes']
['No' 'No']
['yes' 'yes']
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'yes']
['yes' 'yes']
['No' 'No']
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'yes']
['yes' 'yes']
['yes' 'yes']
['yes' 'yes']
['yes' 'yes']
['No' 'No']
['yes' 'yes']
['No' 'No']
['No' 'No']
['yes' 'yes']
```

```
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'No' ]
[ 'yes' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'yes' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
```



```
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'Yes' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'Yes' 'Yes' ]
[ 'Yes' 'Yes' ]
[ 'Yes' 'Yes' ]
```

```
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'yes' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'Yes' ]
[ 'No' 'No' ]
[ 'yes' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'Yes' ]
[ 'No' 'No' ]
[ 'yes' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'yes' 'yes' ]
[ 'yes' 'yes' ]
[ 'No' 'No' ]
[ 'No' 'No' ]]
```

Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, ypred)
print(cm)

[[125 14]
 [ 2 109]]

print("Accuracy =", (125+109)/250*100)

Accuracy = 93.60000000000001

from sklearn.metrics import accuracy_score
print("Test Accuracy =",accuracy_score(y_test,ypred))

Test Accuracy = 0.936
```

Build the Classification Report

```
from sklearn.metrics import classification_report

print(classification_report(y_test, ypred))

precision    recall   f1-score   support

      No      0.98      0.90      0.94      139
     yes      0.89      0.98      0.93      111

  accuracy                           0.94      250
    macro avg      0.94      0.94      0.94      250
weighted avg      0.94      0.94      0.94      250
```

Visualising the Training set results

VISUALIZE TEXT REPRESENTATION

```
!pip install -U scikit-learn

## if not working update scikit learn package
## !pip install -U scikit-learn
from sklearn.tree import export_text
text_representation = export_text(classifier)
print(text_representation)

--- feature_1 <= 1.33
|--- feature_1 <= 0.43
| |--- feature_3 <= 0.27
| | |--- feature_1 <= -0.99
| | | |--- class: yes
| | |--- feature_1 > -0.99
| | | |--- feature_2 <= -0.67
| | | | |--- feature_3 <= -0.08
| | | | | |--- class: yes
| | | | |--- feature_3 > -0.08
| | | | | |--- feature_1 <= 0.19
| | | | | | |--- class: yes
| | | | | |--- feature_1 > 0.19
| | | | | | |--- class: No
| | | | |--- feature_2 > -0.67
| | | | | |--- feature_3 <= -0.36
| | | | | | |--- feature_1 <= -0.51
| | | | | | | |--- class: yes
| | | | | |--- feature_1 > -0.51
| | | | | | |--- feature_2 <= 1.21
| | | | | | | |--- feature_0 <= -0.55
| | | | | | | | |--- class: yes
| | | | | |--- feature_0 > -0.55
```

```
    |--- feature_0 <= 1.26
    |   |--- class: yes
    |--- feature_0 >  1.26
    |   |--- feature_2 <= 0.92
    |       |--- truncated branch of
depth 3
    |--- feature_2 >  0.92
        |--- truncated branch of
depth 2
    |--- feature_2 >  1.21
    |--- feature_1 <= 0.04
        |--- class: yes
    |--- feature_1 >  0.04
        |--- class: No
    |--- feature_3 > -0.36
    |--- feature_1 <= -0.11
        |--- feature_1 <= -0.37
            |--- class: yes
        |--- feature_1 > -0.37
            |--- feature_2 <= 0.73
                |--- class: yes
            |--- feature_2 >  0.73
                |--- feature_0 <= 0.06
                    |--- class: yes
                |--- feature_0 >  0.06
                    |--- class: No
    |--- feature_1 > -0.11
    |--- feature_1 <= 0.15
        |--- feature_0 <= 0.41
            |--- class: No
        |--- feature_0 >  0.41
            |--- feature_2 <= -0.01
                |--- class: yes
            |--- feature_2 > -0.01
                |--- class: No
        |--- feature_1 >  0.15
            |--- class: No
    |--- feature_3 >  0.27
    |--- feature_1 <= -0.66
        |--- feature_1 <= -0.96
            |--- class: yes
        |--- feature_1 > -0.96
            |--- feature_2 <= 1.74
                |--- feature_1 <= -0.88
                    |--- class: yes
                |--- feature_1 > -0.88
                    |--- feature_2 <= -0.39
                        |--- class: yes
                    |--- feature_2 > -0.39
```

```
    |--- feature_3 <= 1.26
    |   |--- class: yes
    |--- feature_3 >  1.26
    |   |--- feature_2 <= 1.01
    |   |   |--- feature_2 <= 0.52
    |   |   |   |--- class: yes
    |   |   |   |--- feature_2 >  0.52
    |   |   |   |   |--- truncated branch of
depth 2
    |--- feature_2 >  1.01
    |   |--- class: No
    |--- feature_2 >  1.74
    |   |--- class: No
    --- feature_1 >  -0.66
    |--- feature_2 <= 0.16
    |   |--- feature_1 <= -0.47
    |   |   |--- class: yes
    |--- feature_1 >  -0.47
    |   |--- feature_3 <= 1.54
    |   |   |--- feature_1 <= 0.28
    |   |   |   |--- feature_1 <= -0.11
    |   |   |   |   |--- feature_1 <= -0.20
    |   |   |   |   |   |--- feature_1 <= -0.24
    |   |   |   |   |   |--- truncated branch of
depth 3
    |--- feature_1 >  -0.24
    |   |--- truncated branch of
depth 3
    |--- feature_1 >  -0.20
    |   |--- feature_1 <= -0.19
    |   |   |--- truncated branch of
depth 2
    |--- feature_1 >  -0.19
    |   |   |--- class: No
    |--- feature_1 >  -0.11
    |   |   |--- class: No
    |--- feature_1 >  0.28
    |   |--- class: No
    --- feature_3 >  1.54
    |--- class: No
    --- feature_2 >  0.16
    |--- feature_2 <= 0.67
    |   |--- feature_0 <= 0.29
    |   |   |--- feature_1 <= -0.42
    |   |   |   |--- class: yes
    |--- feature_1 >  -0.42
    |   |--- feature_0 <= -0.46
    |   |   |--- feature_3 <= 0.88
    |   |   |   |--- class: yes
```

```
          |--- feature_3 >  0.88
          |   |--- class: No
          |--- feature_0 > -0.46
          |   |--- class: No
          |--- feature_0 >  0.29
          |   |--- class: No
          |--- feature_2 >  0.67
          |   |--- class: No
--- feature_1 >  0.43
--- feature_2 <= 0.16
    --- feature_3 <= -0.14
        --- feature_1 <= 0.60
            --- feature_1 <= 0.54
                --- feature_3 <= -0.40
                    |--- class: yes
                --- feature_3 > -0.40
                    |--- class: No
            --- feature_1 >  0.54
                |--- class: yes
--- feature_1 >  0.60
    --- feature_3 <= -1.63
        --- feature_3 <= -1.76
            |--- class: yes
        --- feature_3 > -1.76
            |--- class: No
    --- feature_3 > -1.63
        --- feature_2 <= -0.80
            --- feature_3 <= -1.23
                --- feature_0 <= -0.38
                    |--- feature_0 <= -0.81
                        |--- feature_1 <= 1.22
                            |--- class: yes
                        |--- feature_1 >  1.22
                            |--- class: No
                    --- feature_0 > -0.81
                        |--- class: No
                --- feature_0 > -0.38
                    |--- class: No
                --- feature_3 > -1.23
                    |--- class: No
            --- feature_2 > -0.80
                |--- class: No
        --- feature_3 > -0.14
            |--- class: No
--- feature_2 >  0.16
    |--- class: No
--- feature_1 >  1.33
    |--- class: No
```

Visualize the tree

```
from sklearn import tree
fig = plt.figure(figsize=(25,20))
tree.plot_tree(classifier)
plt.show()
```

