**Topic:**

Building the TN Marginal Workers Assessment with Machine Learnings by feature engineering, model training and evaluation.

**Program:**

**# Data selection & loading**

import pandas

df=pandas.read\_csv("DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv")

print(df.tail())

print(df.head(8))

print(df.info())

print(df.describe())

**# Data cleaning & handling missing values**

print(df.drop\_duplicates())

print(df.dropna())

print(df.isna().any())

print(df.isna().sum())

df.fillna(df.mean(), inplace=True)

cf=df.copy()

**# Feature selection**

x=df[['Age', 'IndustrialCategory', 'Sex']]

y=df['TargetCategory' ]

**# Data splitting**

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)

**# Feature scaling**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x\_train\_scaled = scaler.fit\_transform(x\_train)

x\_test\_scaled = scaler.transform(x\_test)

**# Model training**

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

X = data[['Age', 'IndustrialCategory', 'Sex']]

y = data['TargetCategory']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

**# Evaluation**

# Import necessary libraries

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn import tree

import matplotlib.pyplot as plt

data = pd.read\_csv('data.csv')

X = data[['Age', 'IndustrialCategory', 'Sex']]

y = data['TargetCategory']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

fig = plt.figure(figsize=(12, 8))

\_ = tree.plot\_tree(clf,

feature\_names=['Age', 'IndustrialCategory', 'Sex'],

class\_names=list(map(str, clf.classes\_)),

filled=True)

plt.show()

**Output:**

