# Decision Tree Classifier for Social Network Ads Prediction

## 1. Introduction

This project implements a Decision Tree Classifier to predict whether a user purchases a product based on their demographic details (Gender, Age, Estimated Salary). The dataset used is 'Social\_Network\_Ads.csv'.

## 2. Import Required Libraries

import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.metrics import accuracy\_score, classification\_report

## 3. Load Dataset

df = pd.read\_csv('Social\_Network\_Ads.csv')  
print(df)

## 4. Feature and Target Separation

x = df.drop(['User ID', 'Purchased'], axis=1)  
y = df['Purchased']

## 5. Convert Categorical Data to Numerical

x['Gender'] = x['Gender'].replace({'Male': 0, 'Female': 1})

## 6. Train-Test Split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.25, random\_state=42)

## 7. Model Training

dt\_clf = DecisionTreeClassifier(random\_state=42)  
dt\_clf.fit(x\_train, y\_train)

## 8. Making Predictions

y\_predict = dt\_clf.predict(x\_test)  
print('Prediction', y\_predict)

## 9. Model Evaluation

accuracy = accuracy\_score(y\_test, y\_predict)  
print('Accuracy\_score\_decision\_tree:', accuracy)  
print('classification report:')  
print(classification\_report(y\_test, y\_predict))

## 10. Conclusion

The Decision Tree Classifier achieved high accuracy in predicting purchase decisions based on user demographics. This approach can be further improved by hyperparameter tuning, feature scaling, or using ensemble models like Random Forests.