

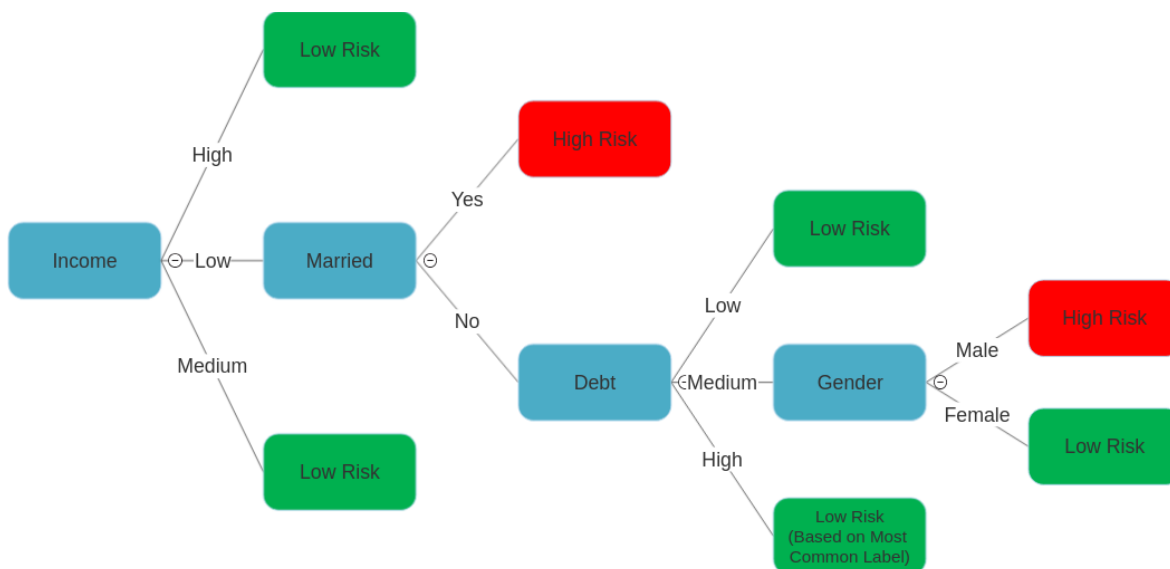
# Homework #1

## **Part 2: Credit Risk Prediction:**

### **Task 2-1:** Draw Tree and compute risk for Tom and Ana

- First, I trained the model I developed with the help of the dataset provided.
- Next, I inferred the Tree from the code and plotted it on an online platform: <https://www.smartdraw.com/decision-tree/>. The tree is shown below(Fig 1)
- Then I calculated accuracy using the same data. I got 100% accuracy (as shown in Fig 2)
- Also, I found the credit risk for Tom and Ana using the same code.
  - Credit Risk for Tom = **LOW**
  - Credit Risk for Ana = **HIGH**

The credit Risk for both can also be inferred from the decision tree drawn below



**Figure 1: Decision Tree**

```
def test_dataset1():  
    """(4.5 points) test dataset1"""  
    X, Y = Tree.load_dataset()  
    t = Tree.train(X, Y)  
    Y_predict = Tree.predict(t, X)  
    #print(Y_predict)  
    accuracy = sum(Y==Y_predict)/9. # training accuracy of 9 training samples  
    print("\n training accuracy:", accuracy)  
  
    X1=['low','low','no','yes','male'] #Data of TOM  
    X2=['low','medium','yes','yes','female'] #Data of ANA  
  
    Y1_predict=Tree.inference(t,X1)  
    Y2_predict=Tree.inference(t,X2)  
    print("\n Credit Risk For:")  
    print("\n 1. TOM = "+str(Y1_predict)) #Printing the Label Predicted for TOM  
    print("\n 2. ANA = "+str(Y2_predict)) #Printing the Label Predicted for ANA  
  
    print("\n \n End of Code")  
    assert accuracy >= 10./9 # Purposely kept accuracy check over 100% to display error
```

```
training accuracy: 1.0  
Credit Risk For:  
1. TOM = low  
2. ANA = low  
End of Code
```

Figure 2: Code O/P

**Task 2-2:** Change Risk for Sofia to “High” and construct a decision tree.

- First, I changed the risk of Sofia to “High” and trained the model based on the new dataset. Below is the Tree that I have plotted using the website mentioned on page 1.

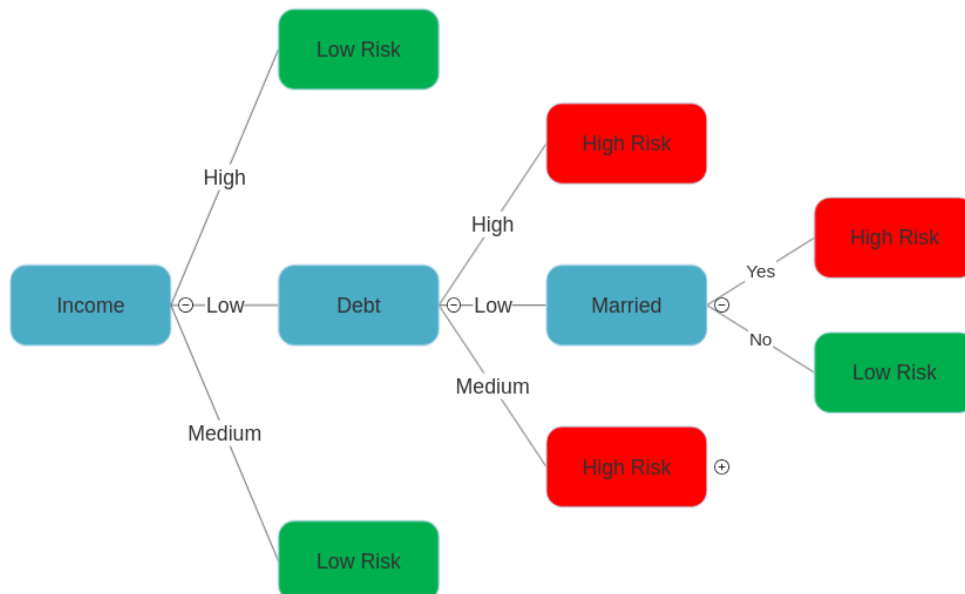


Figure 3: New Decision Tree

- I achieved training accuracy as 100% and Low risk for both TOM and ANA. It can be viewed in the Code Snapshot below (Figure 4)

```
def test_dataset1():
    '''(4.5 points) test dataset1'''
    X, Y = Tree.load_dataset()
    t = Tree.train(X,Y)
    Y_predict = Tree.predict(t,X)
    #print(Y_predict)
    accuracy = sum(Y==Y_predict)/9. # training accuracy of 9 training samples

    print('\n training accuracy:', accuracy)

    X1=['low','low','no','yes','male'] #Data of TOM
    X2=['low','medium','yes','yes','female'] #Data of ANA

    Y1_predict=Tree.inference(t,X1)
    Y2_predict=Tree.inference(t,X2)
    print('\n Credit Risk For:')
    print('\n 1. TOM = '+str(Y1_predict)) #Printing the Label Predicted for TOM
    print('\n 2. ANA = '+str(Y2_predict)) #Printing the Label Predicted for ANA

    print('\n \n End of Code')
    assert accuracy >= 10./9 # Purposely kept accuracy check over 100% to display
```

```
1 Name,Risk,Debt,Income,Married?,Owns_Property,Gender
2 Tim,low,low,low,no,no,male
3 Joe,low,high,high,yes,yes,male
4 Sue,low,low,high,yes,no,female
5 John,high,medium,low,no,no,male
6 Mary,high,high,low,yes,no,female
7 Fred,high,low,low,yes,no,male
8 Pete,low,low,medium,no,yes,male
9 Jacob,low,high,medium,yes,yes,male
10 Sofia,high,medium,low,no,no,female
```

```
training accuracy: 1.0
Credit Risk For:
1. TOM = low
2. ANA = low

End of Code
----- >> end captured stdout << -----
Ran 14 tests in 0.010s
```

Figure 4: Code O/p

- Features not playing a role in the first decision tree is: **“Own\_Property”**