CSC-411 Artificial Intelligence

Introduction to Machine Learning

Classification

Classification

- The problem of discriminating between different classes of objects
- Classification process:
  - Find examples for which you know the class (training set)
  - Find a set of features that discriminate between the examples within the class and outside the class
  - Create a function that given the features decides the class
  - Apply the function to new examples.

Catching Tax Fraud

Tax-return data for year 2019

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

A new tax return for 2020
Is this a cheating tax return?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

An instance of the classification problem: learn a method for discriminating between records of different classes (cheaters vs. non-cheaters)

What is Classification?

- **Classification** is the task of ***learning a target function f*** that maps attribute set **x** to one of the predefined class labels **y**

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

One of the attributes is the class attribute: In this case: Cheat

Two class labels (or classes): Yes (1), No (0)

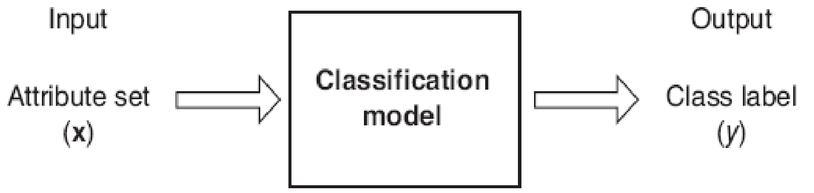


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

Why Classification?

- The target function f is known as a classification model

- Descriptive modeling: Explanatory tool to distinguish between objects of different classes (e.g., understand why people cheat on their taxes, or what makes a hipster)

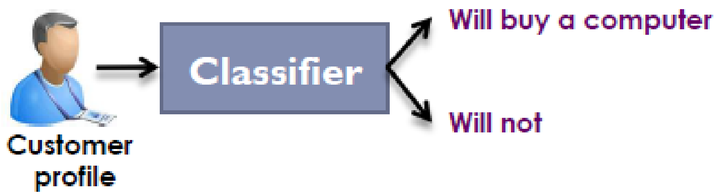
- Predictive modeling: Predict a class of a previously unseen record

Examples of Classification Tasks

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Categorizing news stories as finance, weather, entertainment, sports, etc.
- Identifying spam email, spam web pages, adult content
- Understanding if a web query has commercial intent or not

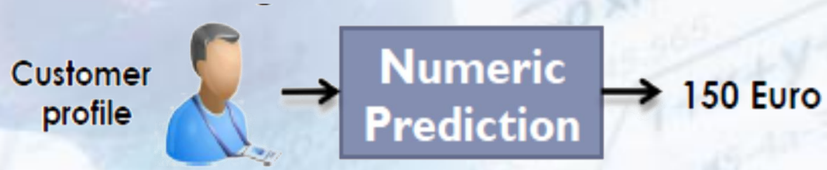
Classification vs. Prediction

- Classification
  - Predicts categorical class labels (discrete or nominal)
  - Use labels of the training data to classify new data
- Example



  - A model or classifier is constructed to predict categorical labels such as safe" or "risky" for a loan application data.

- Prediction
  - Models continuous-valued functions, i.e., predicts unknown or missing values
- Example
  - A marketing manager would like to predict how much a given costumer will spend during a sale.



  - Unlike classification, it provides ordered values
  - Regression analysis is used for prediction
  - Prediction is a short name for numeric prediction

General approach to Classification

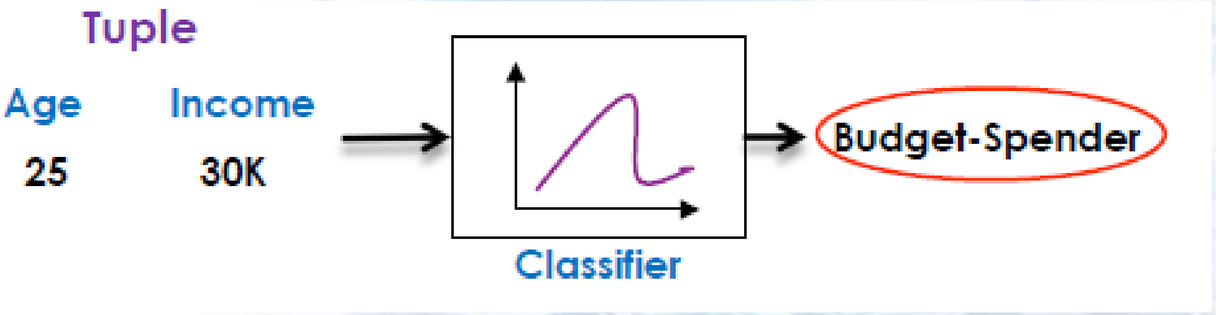
- Training set consists of records with known class labels
- Training set is used to build a classification model
- A labeled test set of previously unseen data records is used to evaluate the quality of the model.
- The classification model is applied to new records with unknown class labels.
- Accuracy rate is the percentage of test set samples that are correctly classified by the model
- Important: test data should be independent of training set, otherwise over-fitting will occur

General approach to Classification

- Before using the model, we first need to test its accuracy
  - Measuring model accuracy
    - To measure the accuracy of a model we need test data
    - Test data is similar in its structure to training data (labeled data)
    - How to test?
      - The known label of test sample is compared with the classified result from the model

Test data

Age	Income	Class label
25	30K	Budget-Spenders
40	50k	Big-Spender



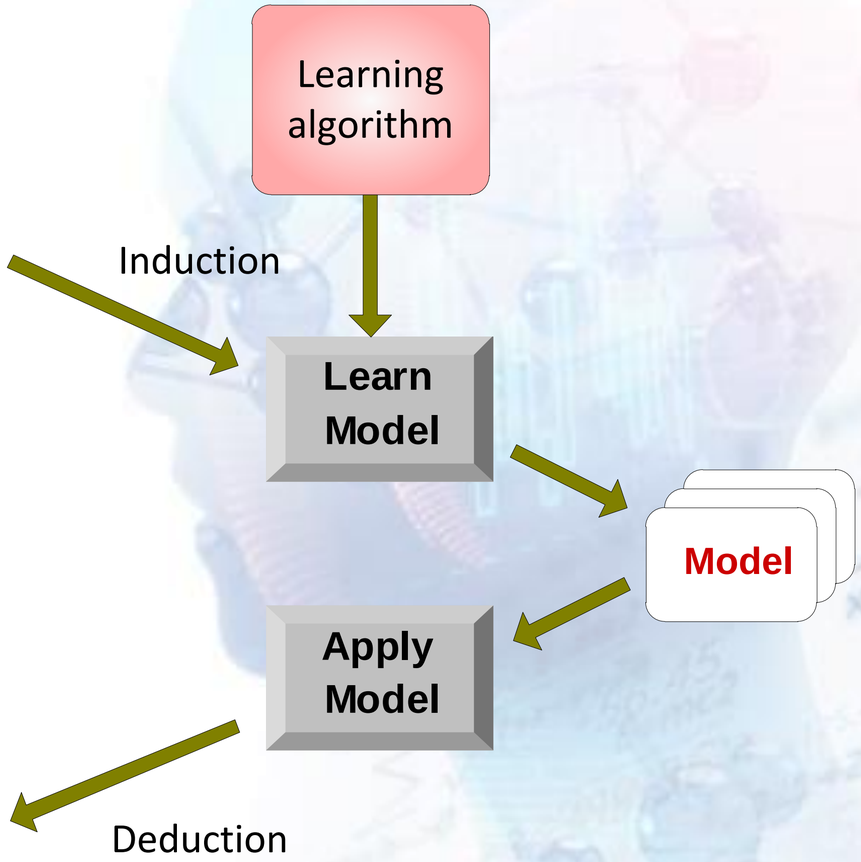
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

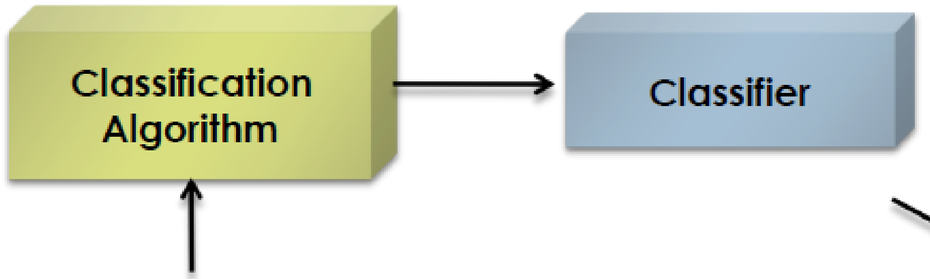
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Model Construction



If age <30 & income <30k
Then, Budget-Spender

If age <30 & income > 30k
Then, Big-Spender

If 30<age<60 and income >30k
Then Big-spender

If 30<age<60 and income <30k
Then Budget-spender

If age >60, Then Budget-spender

Training data

Age	Income	Class label
27	28K	Budget-Spender
35	36K	Big-Spender
38	28K	Budget-Spender
65	45K	Budget-Spender
20	18k	Budget-Spender
75	40k	Budget-Spender
28	50k	Big-Spender
40	60k	Big-Spender
60	65k	Big-Spender

Evaluation of Classification Models

- Counts of test records that are correctly (or incorrectly) predicted by the classification model
- Confusion matrix

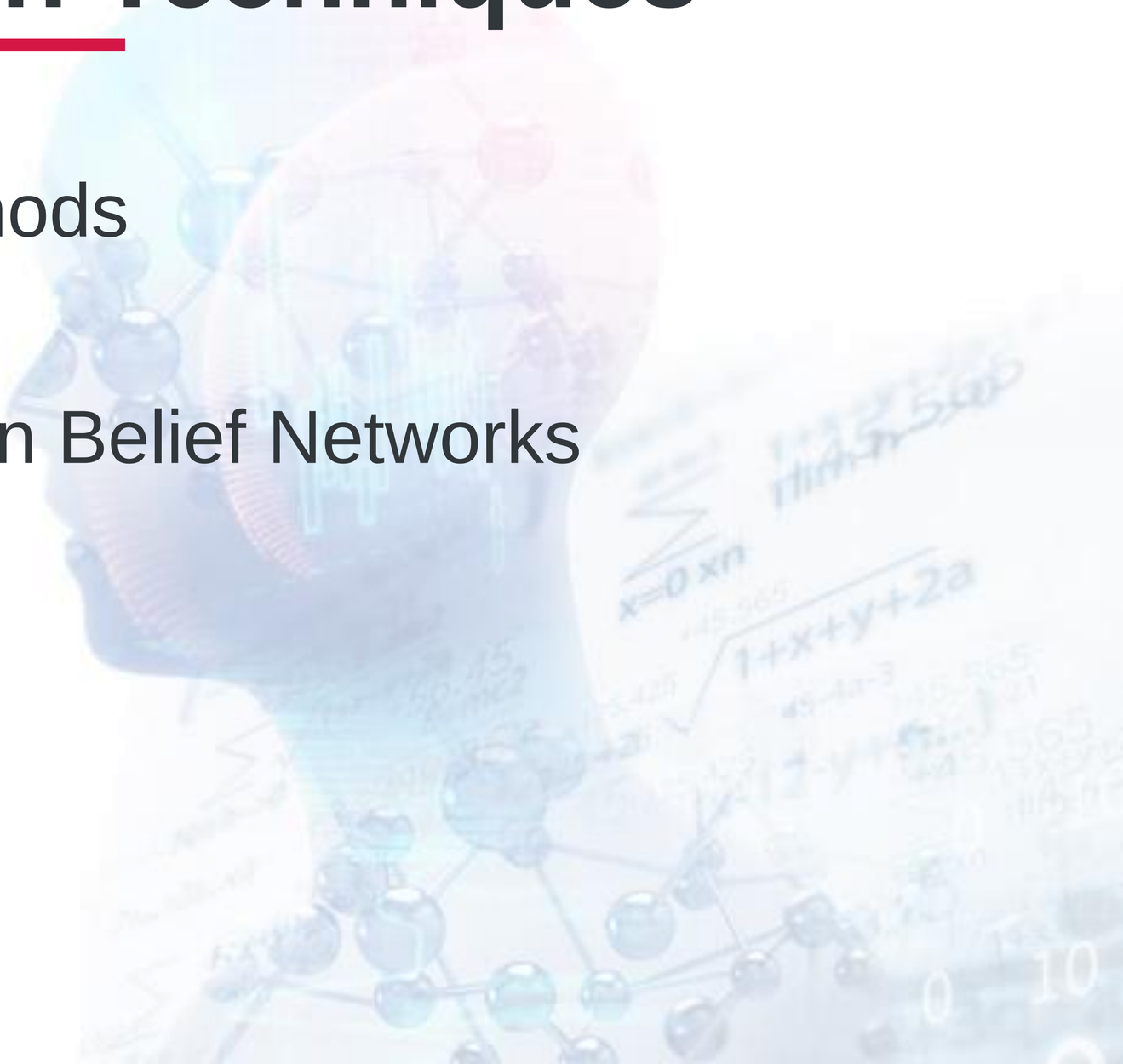
Actual Class	Predicted Class: Class = 1	Predicted Class: Class = 0
Class = 1	(TP) f_{11}	(FP) f_{10}
Class = 0	(FN) f_{01}	(TN) f_{00}

$$\text{Accuracy} = \frac{\#\,\text{correct predictions}}{\text{total}\,\#\,\text{of predictions}} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{Error rate} = \frac{\#\,\text{wrong predictions}}{\text{total}\,\#\,\text{of predictions}} = \frac{FP + FN}{TP + FP + FN + TN}$$

Classification Techniques

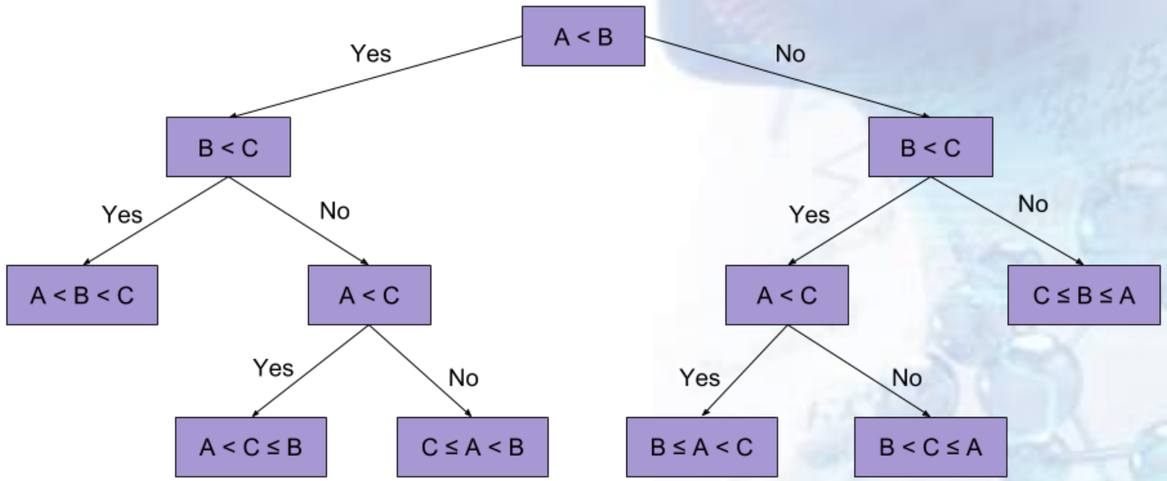
- Decision Tree based Methods
- Rule-based Methods
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines
- Neural Networks



Decision Trees

Decision Trees

- Decision Tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution

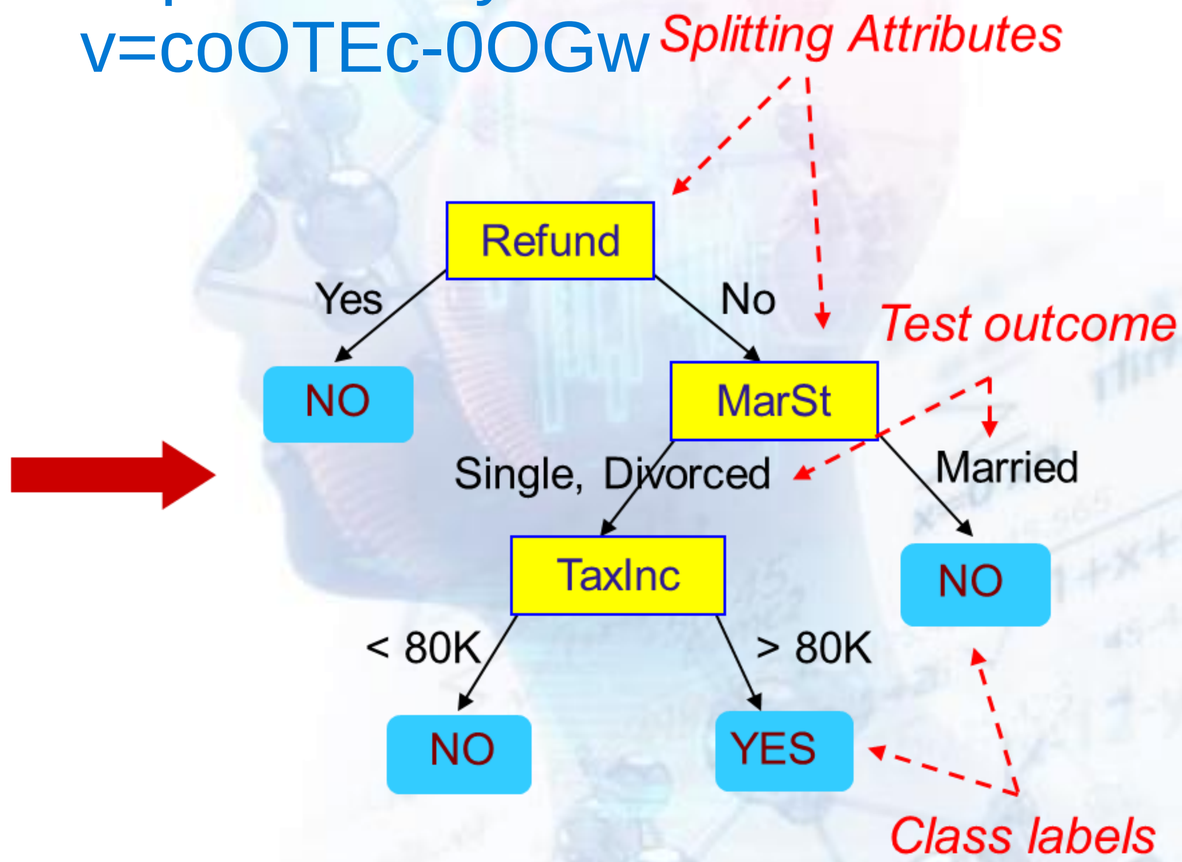


Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
		categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

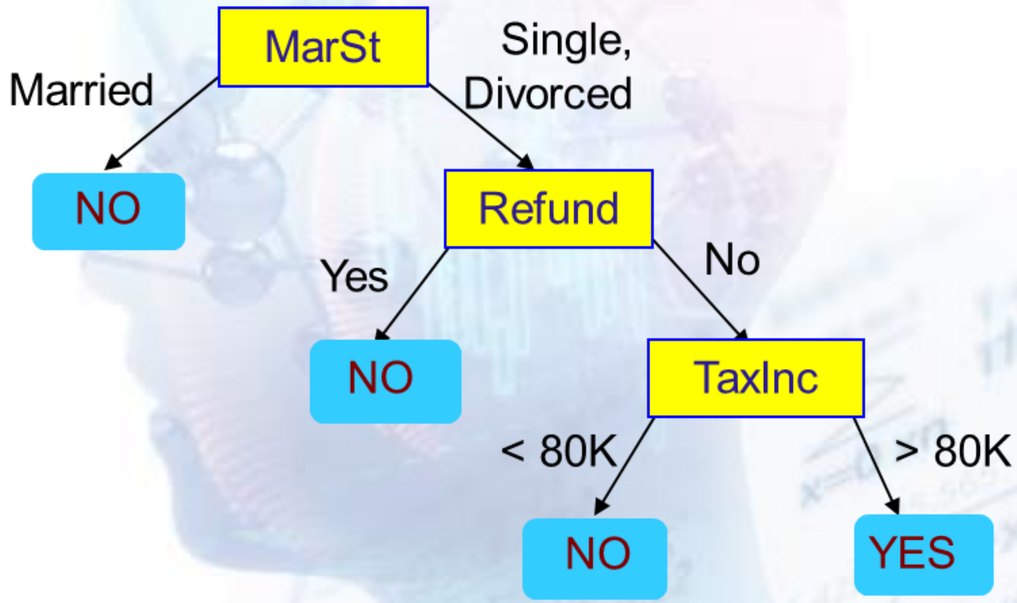
https://www.youtube.com/watch?v=coOTEc-0OGw



Model: Decision Tree

Another Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



There could be more than one tree that fits the same data!

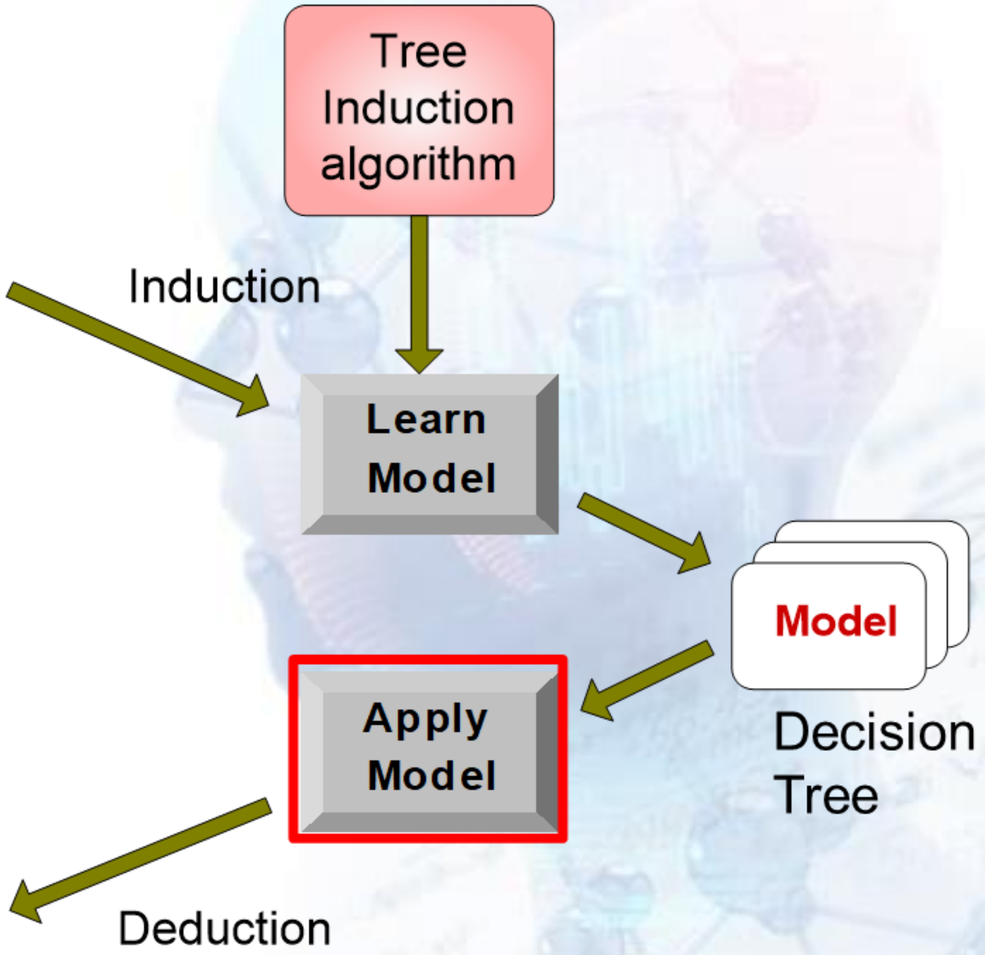
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

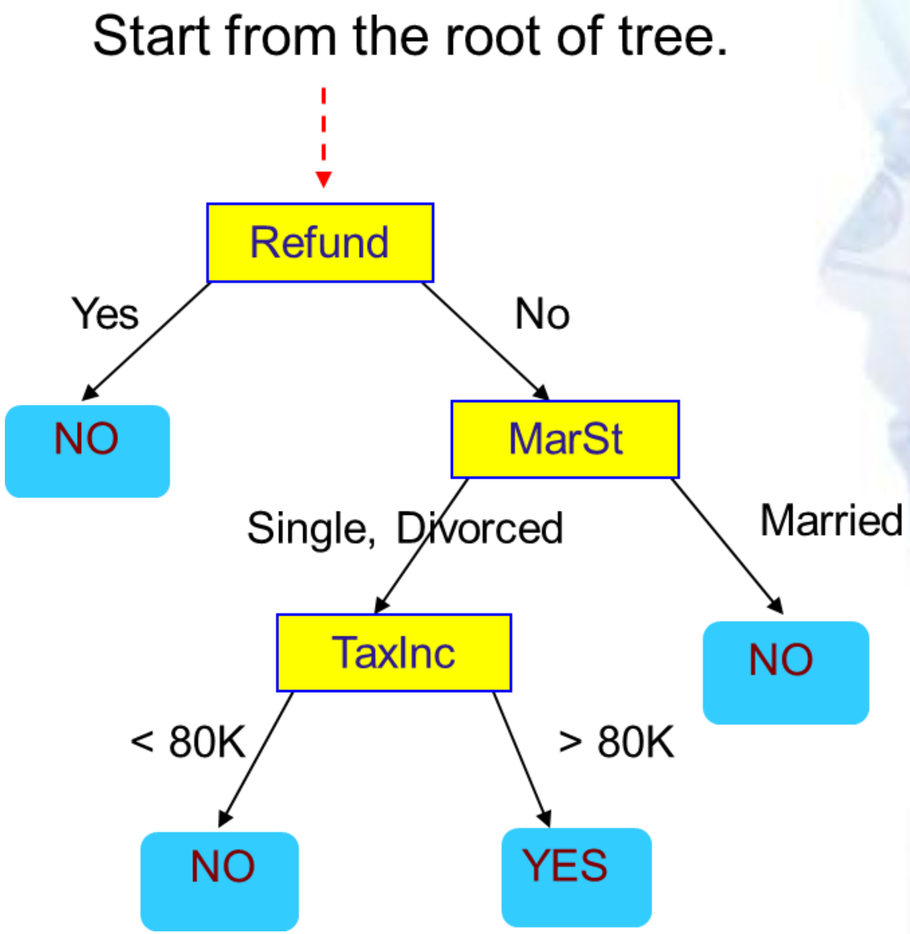
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Apply Model to Test Data



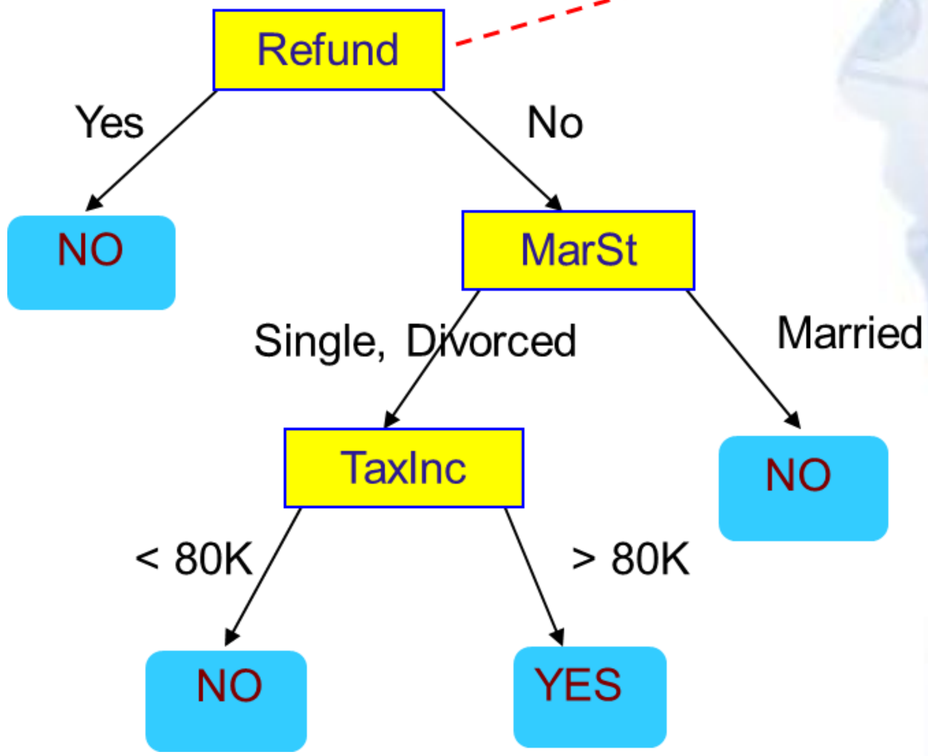
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

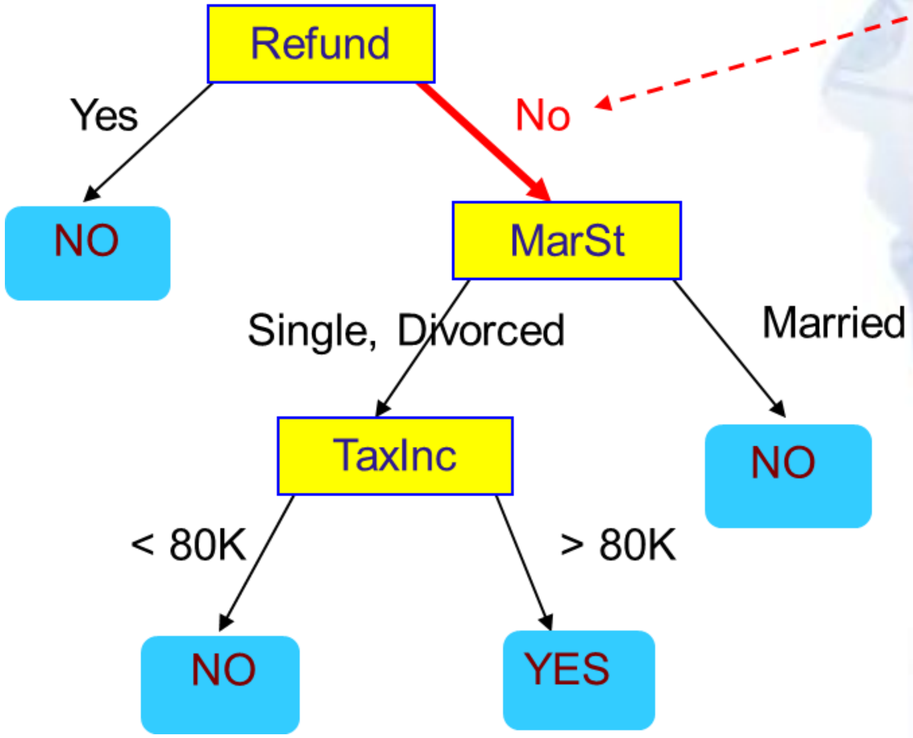
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



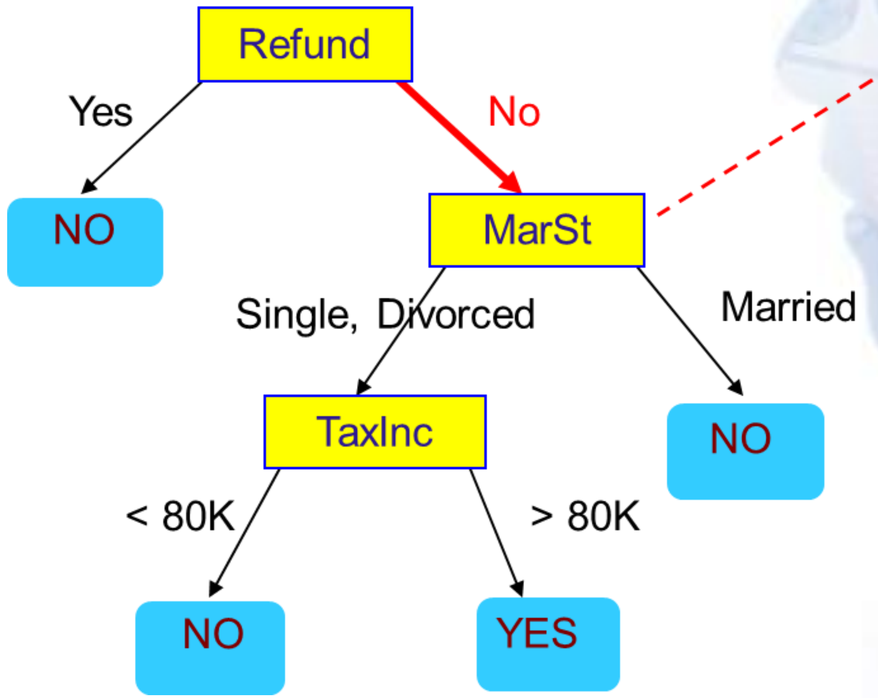
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



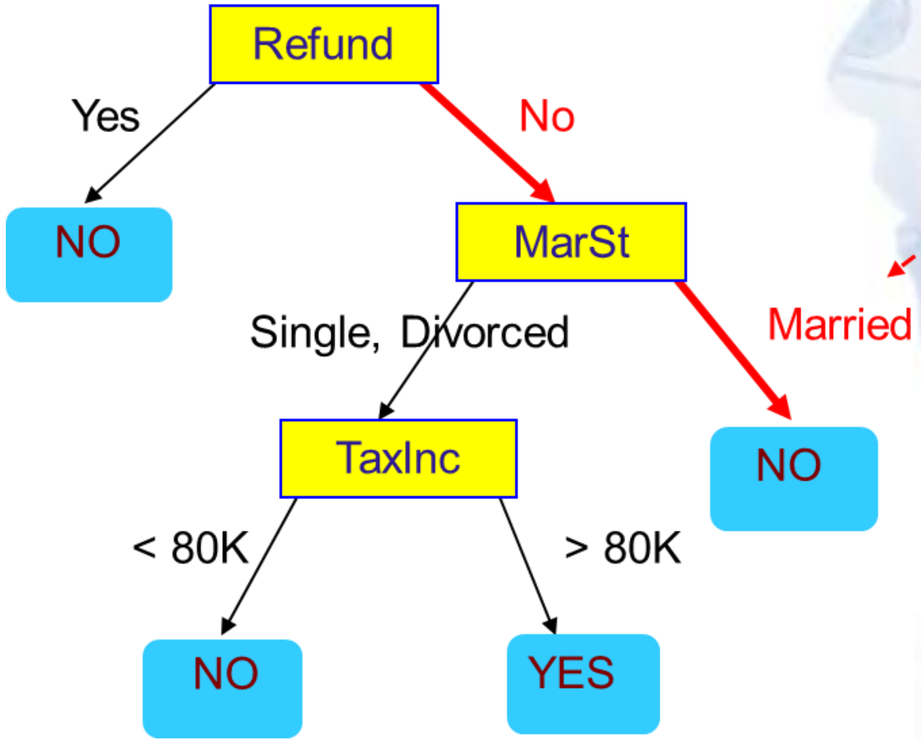
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

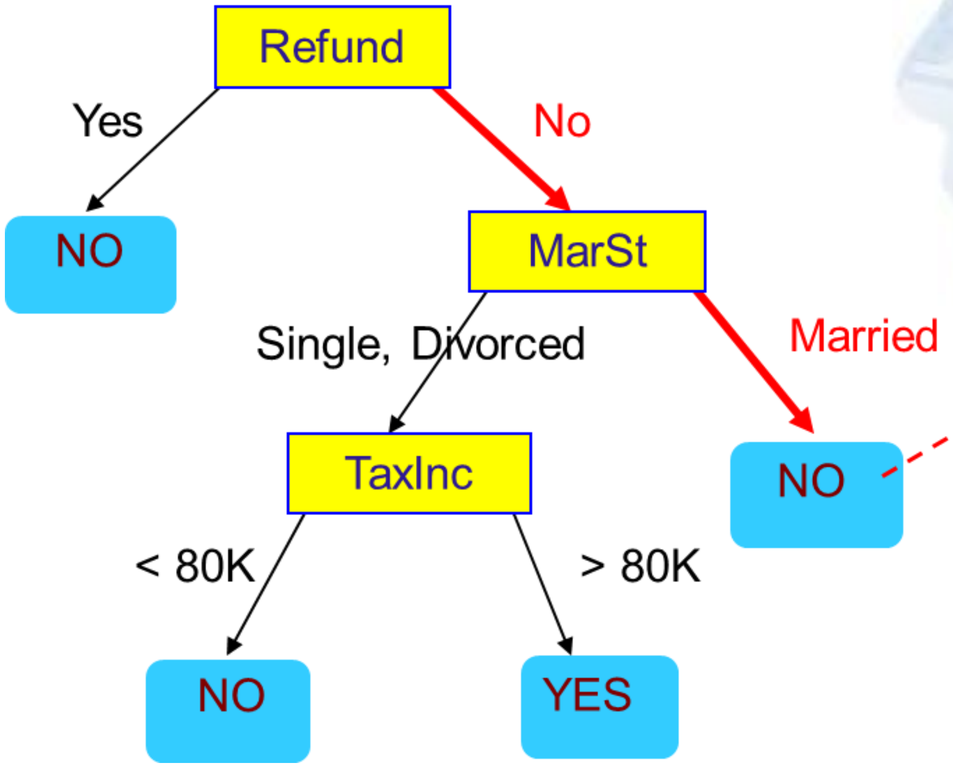
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



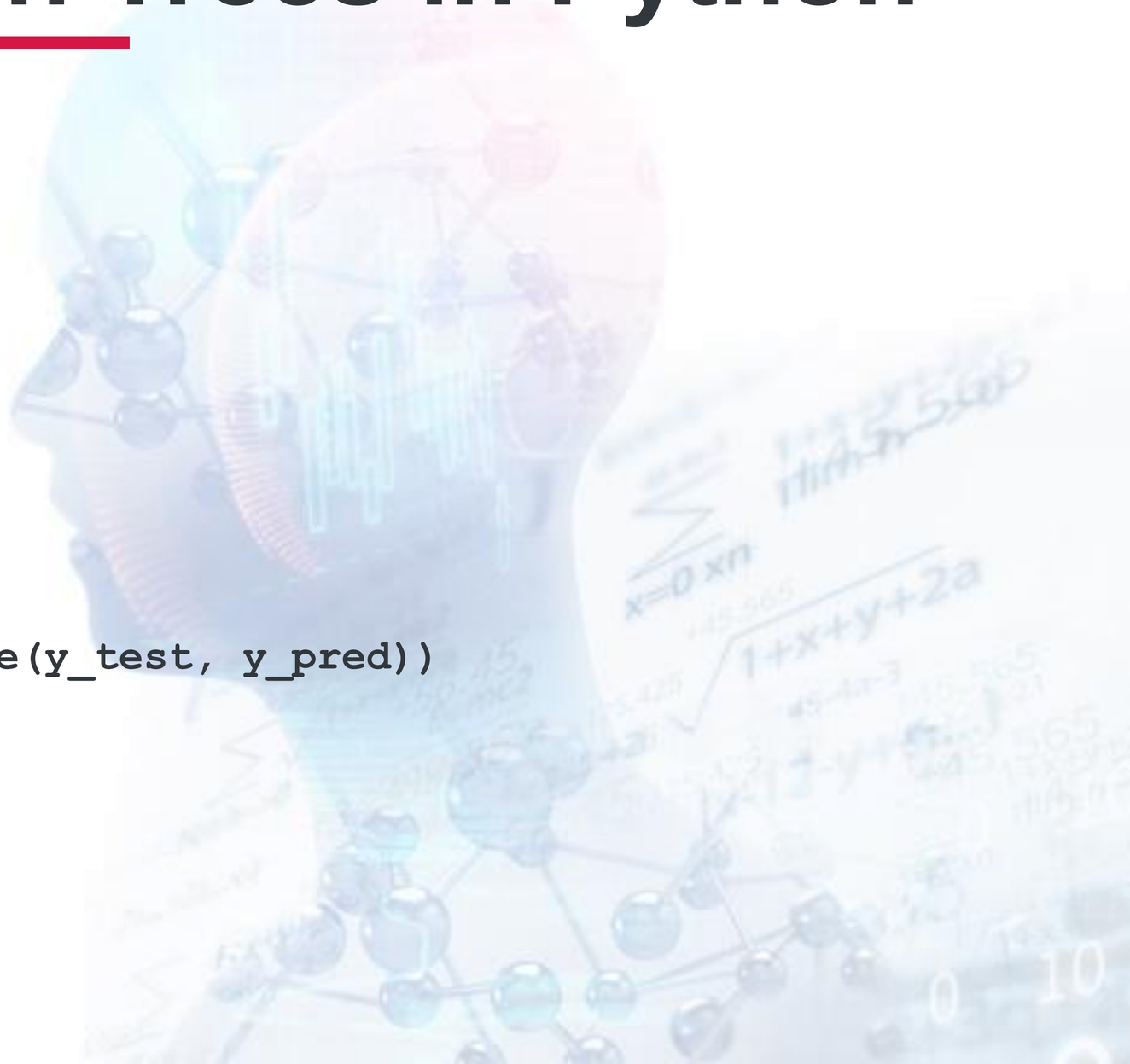
Assign Cheat to “No”

Building Decision Trees in Python

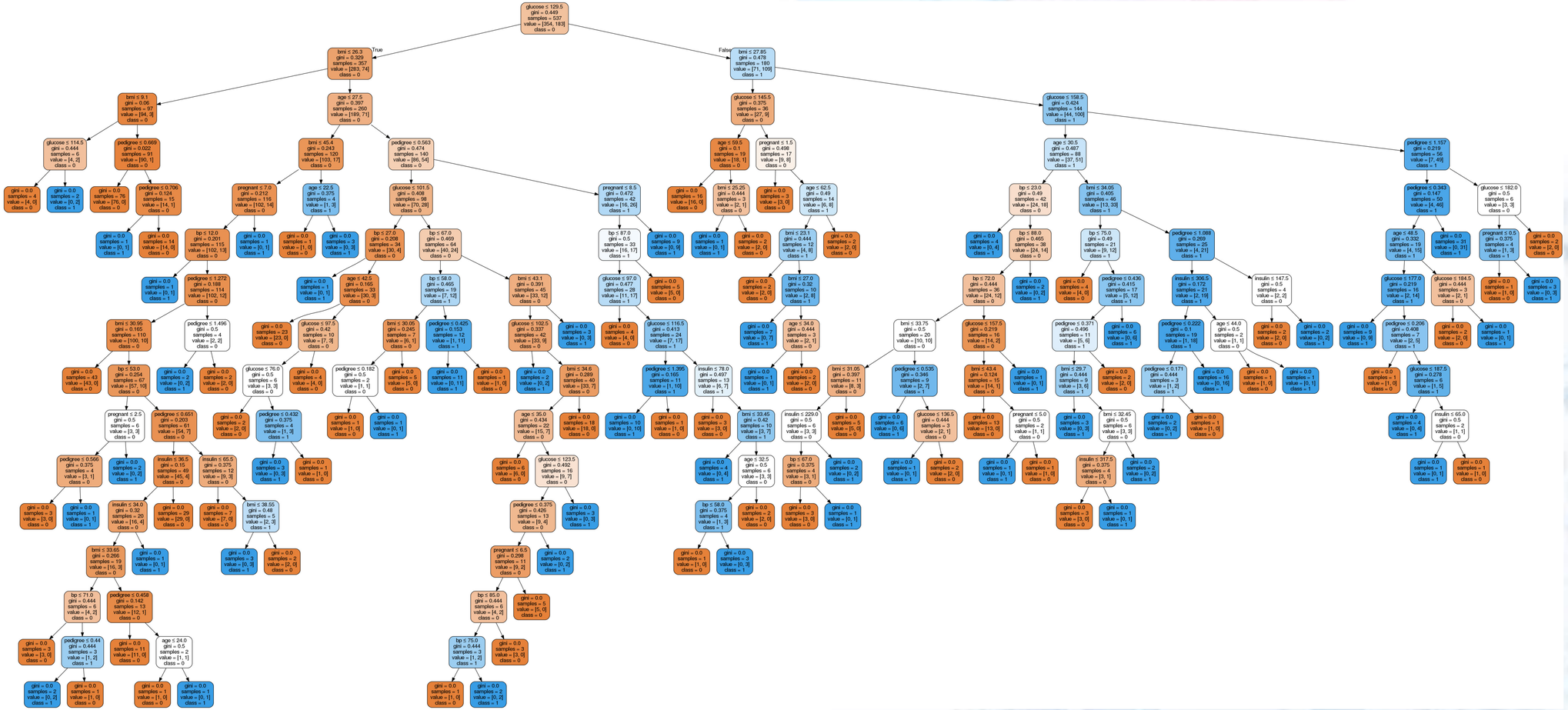
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics

col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree',
'age', 'label']
pima = pd.read_csv("pima-diabetes.csv", header=None, names=col_names)

# Split dataset in features and target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols] # Features
y = pima.label # Target variable

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1) # 70% training and 30% test
```

Building Decision Trees in Python

```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```



Building Decision Trees in Python



Pruned Decision Trees in Python

```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Pruned Decision Trees in Python