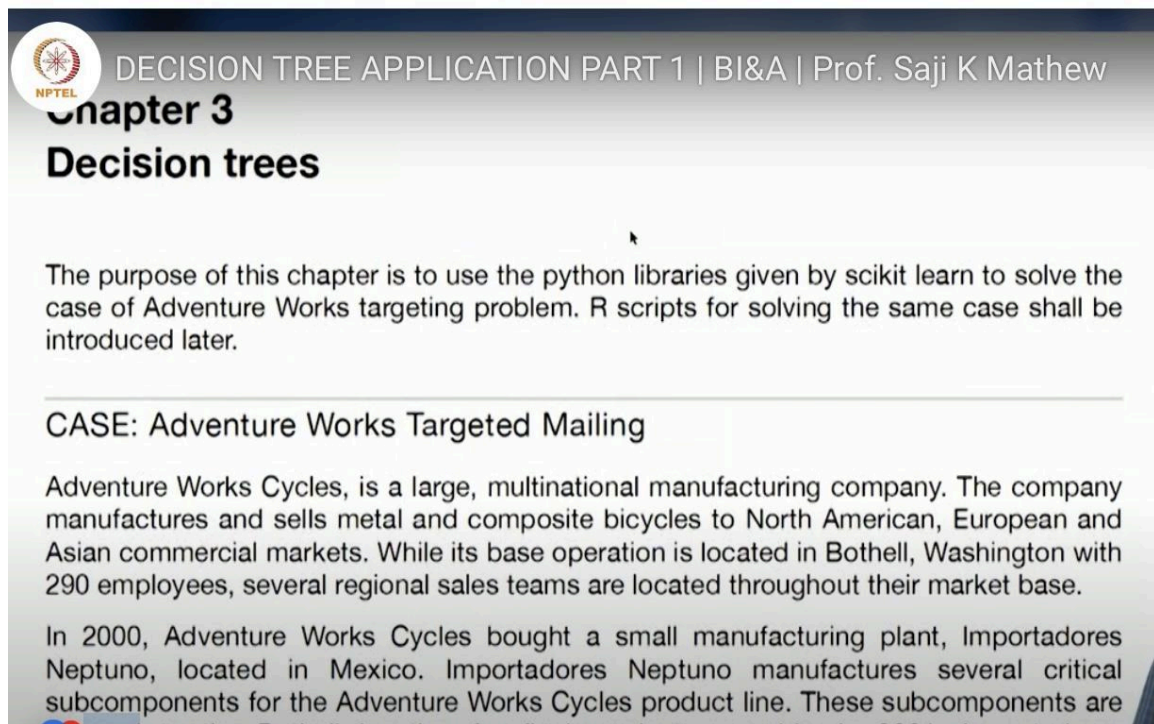


Course Name:Business Intelligence and Analytics
Professor Name:Prof. Saji.K.Mathew
Department Name:Department of Management Studies
Institute Name:Indian Institute of Technology Madras
Week:08
Lecture:28

DECISION TREE APPLICATION PART 1

Okay, good afternoon and welcome back to this course. Today's session is an extension of decision trees which we discussed in the previous session. Decision tree is a supervised learning technique which we discussed sufficiently as a part of understanding classification. So decision tree can be used to build classifiers and then you can test it for its performance and use it for solving problems. So the aim of today's class is to discuss an application of decision trees, which is a part of the readings that is given to you. So you can look at the description of the case that we are going to use today for decision trees.



DECISION TREE APPLICATION PART 1 | BI&A | Prof. Saji K Mathew

Chapter 3
Decision trees

The purpose of this chapter is to use the python libraries given by scikit learn to solve the case of Adventure Works targeting problem. R scripts for solving the same case shall be introduced later.

CASE: Adventure Works Targeted Mailing

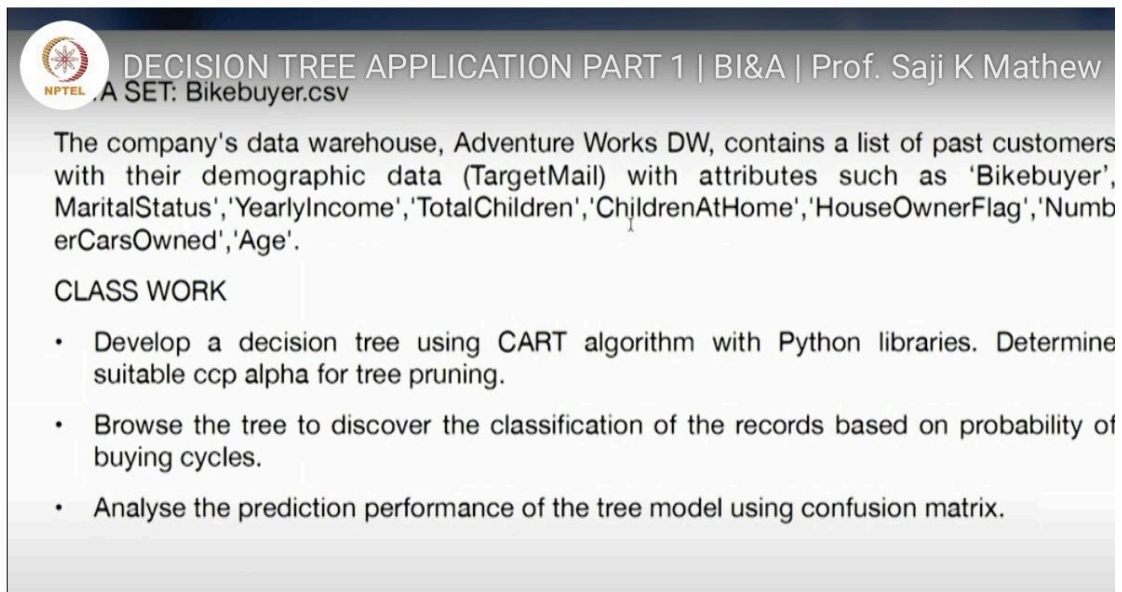
Adventure Works Cycles, is a large, multinational manufacturing company. The company manufactures and sells metal and composite bicycles to North American, European and Asian commercial markets. While its base operation is located in Bothell, Washington with 290 employees, several regional sales teams are located throughout their market base.

In 2000, Adventure Works Cycles bought a small manufacturing plant, Importadores Neptuno, located in Mexico. Importadores Neptuno manufactures several critical subcomponents for the Adventure Works Cycles product line. These subcomponents are

The case is titled Adventure Works Targeted Mailing. So the case talks about a cycle manufacturing company or bicycle manufacturing company, in the west it is called bike. So when we hear the term bike, we generally expect that it is a motorbike but that is not

the case if you go to Europe or America. So bike is a cycle for us and this company actually is a manufacturing company and it also has distribution networks and marketing channels and it also sells online.

So and this company also has developed analytics as a practice and therefore it has built infrastructure, which we discussed like data warehouse and tables which capture relevant data from databases and store in the data warehouse for data analysis purpose. So one of the files that they have captured in the data warehouse is the bike buyer table. It is a table basically, the bike buyer table I have downloaded from the database of this company, this fictitious company and converted that into a CSV file. So in the CSV file you will find if you open bike buyer. CSV. So I suggest that you open this file bike buyer.CSV which is there in the data folder.



NPTEL **DECISION TREE APPLICATION PART 1 | BI&A | Prof. Saji K Mathew**
DATA SET: Bikebuyer.csv

The company's data warehouse, Adventure Works DW, contains a list of past customers with their demographic data (TargetMail) with attributes such as 'Bikebuyer', 'MaritalStatus', 'YearlyIncome', 'TotalChildren', 'ChildrenAtHome', 'HouseOwnerFlag', 'NumberCarsOwned', 'Age'.

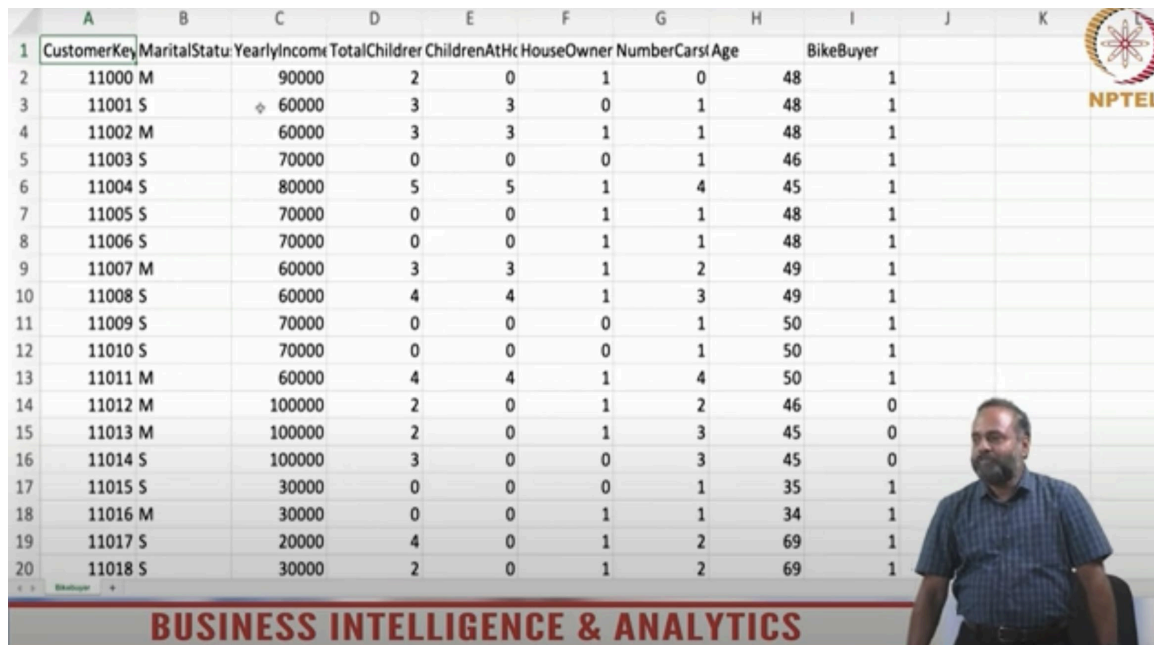
CLASS WORK

- Develop a decision tree using CART algorithm with Python libraries. Determine suitable ccp alpha for tree pruning.
- Browse the tree to discover the classification of the records based on probability of buying cycles.
- Analyse the prediction performance of the tree model using confusion matrix.

So this is the data source or this is the data that we are going to use today. So what is given to us is a targeting problem. A problem which we have been discussing a lot without any data but today we are going to use some data and model it. So in targeting, the main decision is, well you have a larger data set of potential customers but you do not have all the money to go after everyone. It is not required, you know you have to be very targeted or focused in your approach so that you spend your money and time wisely.

So therefore from the larger database you decide whom to target based on your budget, that is the targeting problem. So, and a target whom you target or whom you reach out to should be a relevant or responding target, a target who is likely to respond. So if you reach out to people who are not going to buy your product, you are wasting your resources. So that is the problem in targeting, whom should you target, whom should you

reach out to. So what we have to take this decision is a lot of attributes about the various target, potential targets or the different elements in the database or different prospects in the database, you have their characteristics as well it is not just their address or name but certain attributes of the customers, potential customers are also given.



| CustomerKey | MaritalStatus | YearlyIncome | TotalChildren | ChildrenAtHome | HouseOwner | NumberCars | Age | BikeBuyer |
|-------------|---------------|--------------|---------------|----------------|------------|------------|-----|-----------|
| 11000 | M | 90000 | 2 | 0 | 1 | 0 | 48 | 1 |
| 11001 | S | 60000 | 3 | 3 | 0 | 1 | 48 | 1 |
| 11002 | M | 60000 | 3 | 3 | 1 | 1 | 48 | 1 |
| 11003 | S | 70000 | 0 | 0 | 0 | 1 | 46 | 1 |
| 11004 | S | 80000 | 5 | 5 | 1 | 4 | 45 | 1 |
| 11005 | S | 70000 | 0 | 0 | 1 | 1 | 48 | 1 |
| 11006 | S | 70000 | 0 | 0 | 1 | 1 | 48 | 1 |
| 11007 | M | 60000 | 3 | 3 | 1 | 2 | 49 | 1 |
| 11008 | S | 60000 | 4 | 4 | 1 | 3 | 49 | 1 |
| 11009 | S | 70000 | 0 | 0 | 0 | 1 | 50 | 1 |
| 11010 | S | 70000 | 0 | 0 | 0 | 1 | 50 | 1 |
| 11011 | M | 60000 | 4 | 4 | 1 | 4 | 50 | 1 |
| 11012 | M | 100000 | 2 | 0 | 1 | 2 | 46 | 0 |
| 11013 | M | 100000 | 2 | 0 | 1 | 3 | 45 | 0 |
| 11014 | S | 100000 | 3 | 0 | 0 | 3 | 45 | 0 |
| 11015 | S | 30000 | 0 | 0 | 0 | 1 | 35 | 1 |
| 11016 | M | 30000 | 0 | 0 | 1 | 1 | 34 | 1 |
| 11017 | S | 20000 | 4 | 0 | 1 | 2 | 69 | 1 |
| 11018 | S | 30000 | 2 | 0 | 1 | 2 | 69 | 1 |

BUSINESS INTELLIGENCE & ANALYTICS

So here for example if you look at the adventure works database, you can see that each record has an identifier or a primary key which is the customer key and then certain attributes of that particular customer. So what are those attributes, it talks about the marital status, you can see single M, S or M. So looks like a binary, nothing in between of course and the second attribute is yearly income, you can see it is a, it is a numeric data and total children at home. So you can see it is a number, it is an integer, it is a count and next attribute is children, total children that is one attribute, children at home both are different, right. So both the attributes are captured house owner, in the sense whether the particular prospect owns a house or not and number of cars at home age.

So when we discussed data warehouse, we and when we discussed the bizocity score as a problem, we recognize that these are problems related to variable selection. you select certain variables for a problem. So you select variables such that ,one data about those variables are available and number two, they are good predictors of a certain behavior certain outcome which you want to model and here the outcome is what, there is a last column which talks about bike buyer. So bike buyer is a binary variable, you can see 0 or 1. 1 means this particular person has bought a cycle in the past and 0 means this person has not bought a cycle. Therefore the variable of interest or the target variable is

the bike buyer variable and it has two values or two labels or two classes, all are the same.

This is a two class variable, 1 and 0. 1 is a buyer, 0 is a non buyer and that is what we are interested in. So here in this case, we are interested to know what rules determine a 1 or what conditions lead to an outcome which is 1 and that is something that you want to use to predict future potential buyers. So this is historical data, but from this if you have a new set of customer data that you obtained, you want to target them based on the rules that you will develop from this data, that is what we are doing. So a decision tree help us discover the rules and then we apply that to a new set of data, that is prediction.

So that is the problem that we are going to address in today's class. So in order to do this, we are going to employ Python programming. So I am sure you are all trained in Python to some extent and if not you are getting trained and you are soiling your hands and having some fun, I believe. So you started with Hello World and I am sure I have referred a textbook to you, which is very much a textbook related to Python for data science. This McKinney's book is something I strongly recommend I use that book to learn Python for data science myself.

So, please do refer and sit with your programming platform, whether Anaconda or Google, whatever you want to use you can use Google platform for the same purpose, I will be using Anaconda. So let me actually open Anaconda Navigator and the Jupyter notebook, they help us run the code from the browser. Of course, the Jupyter notebook is an interface to write the codes and you write it cell wise or it is very modular. So you can put, write related instructions together and execute it.

So the first effort is to call the relevant libraries or to invoke the libraries so that the functions in the libraries can be called. So the first step is that I am invoking certain libraries which are already existing. So which is the library that I am extensively using here, in addition to the generic instructions that are available in Python. I am importing a set of functions from a library known as scikit-learn or sklearn. Sklearn is a library which was developed, I guess by Google.

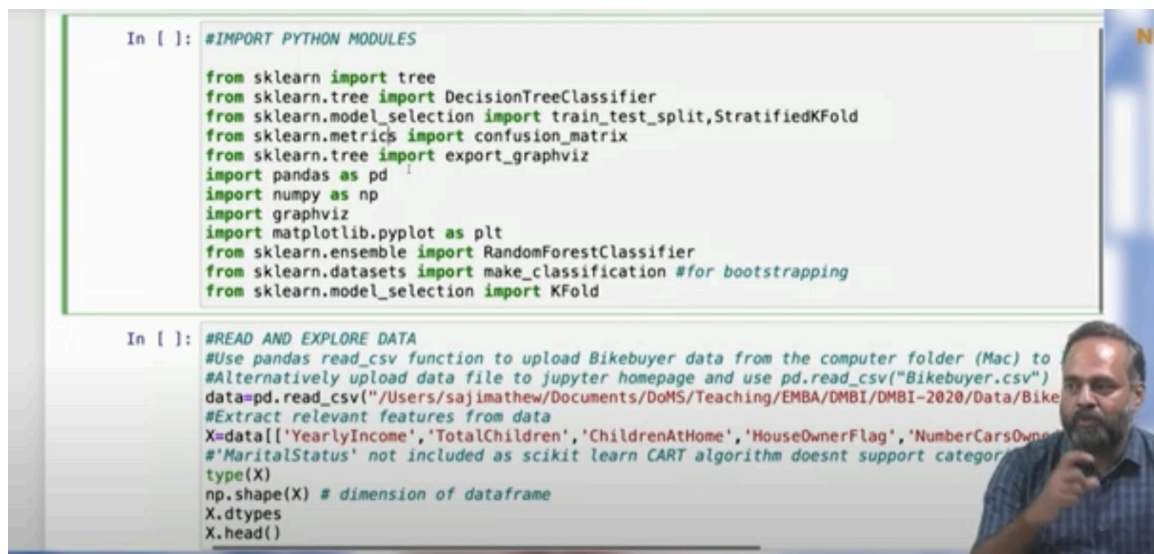
It was a summer project of certain some students. So it is an open source. It is an open code which has several useful functions of machine learning. So it has already coded machine learning algorithms and functions and you can directly import them and use them. So sklearn means scikit-learn and scikit-learn library also comes with extensive documentation.

In your assignment, you must be referring to scikit-learn documentation and I have

given you a specific references for that. So we are going to use that. The first thing I am doing is, I am importing the tree because we are going to build decision tree. The first one would be the CART algorithm and I am also using decision tree classifier because I want to look at how the decision tree has performed with the test data. So that is a separate function I am actually importing.

Then I am importing train test split. That is actually a function available in the scikit-learn model selection. So this is actually for splitting the data set into train data and test data, we discussed about it. And stratified k-fold is the, it gives the function for k-fold cross validation. And then there is another library scikit-learn.metrics and the confusion matrix or different measures that are used to evaluate classifiers, that is all developed in a library known as metrics. So confusion matrix is one of them. So I am importing that. Export graphviz is actually, this is useful for visualizing the graph. And then in addition to scikit-learn functions, it is all mixed up here when I call them.

I am using pandas. I hope you are learning pandas already, you have learned pandas already. Numpy and pandas. Numpy basically for arrays or algebraic operations. Pandas for the data frames. You can actually work with tables of different columns having different data types.



```
In [ ]: #IMPORT PYTHON MODULES

from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.metrics import confusion_matrix
from sklearn.tree import export_graphviz
import pandas as pd
import numpy as np
import graphviz
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification #for bootstrapping
from sklearn.model_selection import KFold

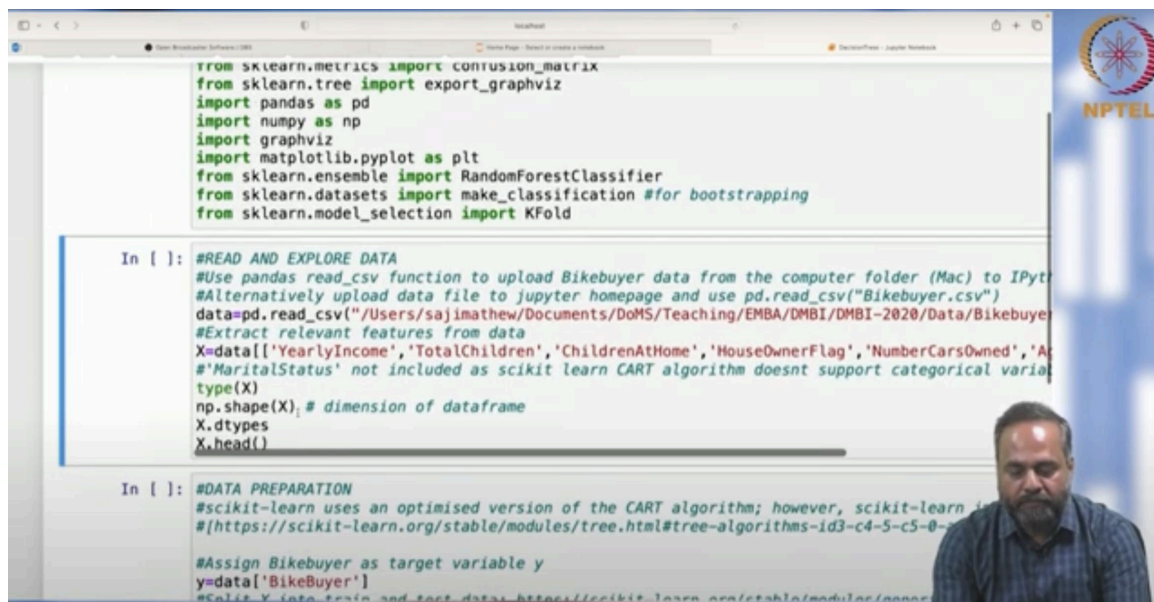
In [ ]: #READ AND EXPLORE DATA
#Use pandas read_csv function to upload Bikebuyer data from the computer folder (Mac) to
#Alternatively upload data file to jupyter homepage and use pd.read_csv("Bikebuyer.csv")
data=pd.read_csv("/Users/sajimathew/Documents/DoMS/Teaching/EMBA/DMBI/DMBI-2020/Data/Bike
#Extract relevant features from data
X=data[['YearlyIncome', 'TotalChildren', 'ChildrenAtHome', 'HouseOwnerFlag', 'NumberCarsOwned',
# 'MaritalStatus' not included as scikit learn CART algorithm doesnt support categorical
type(X)
np.shape(X) # dimension of dataframe
X.dtypes
X.head()
```

That is a data frame. So if you have to call a table data with multiple data types and work on them, you need a data frame. Pandas provide that. Numpy and graphviz as I said is for visualization and matplotlib is also for visualization. So graphviz is developed specifically for decision tree visualization and matplotlib is a very generic library for different types of graphs. And I am also calling functions for random forest classifier and a function for bootstrapping.

You can see my classification and k-fold is already done. So let me execute this depending on whether you are using Windows or Mac, you have shortcuts, different shortcuts for execution. Of course at the top you find a run. So when you are in a particular cell, you will be running the codes in that cell.

So all the codes are fine. So all these libraries and their corresponding functions have been imported to the Jupyter notebook platform and we are going to use those functions now. Now in the second cell, yes, you have to install it separately. You have to install it separately, the procedure is given in your course outline. That has to be installed separately. It will not come if you have not done, you had run some code to do that.

Now, next is we are going to read the data. In our case, the data pertains to a problem of bike buyer. So targeting bike buyers, that is our problem. We looked at the data already. So you can read the data from the folder where you have stored that data by providing the path here, that is one way or if you upload the data into the Jupyter folder, which we saw previously, you do not have to give all this path but you can just use the file name.



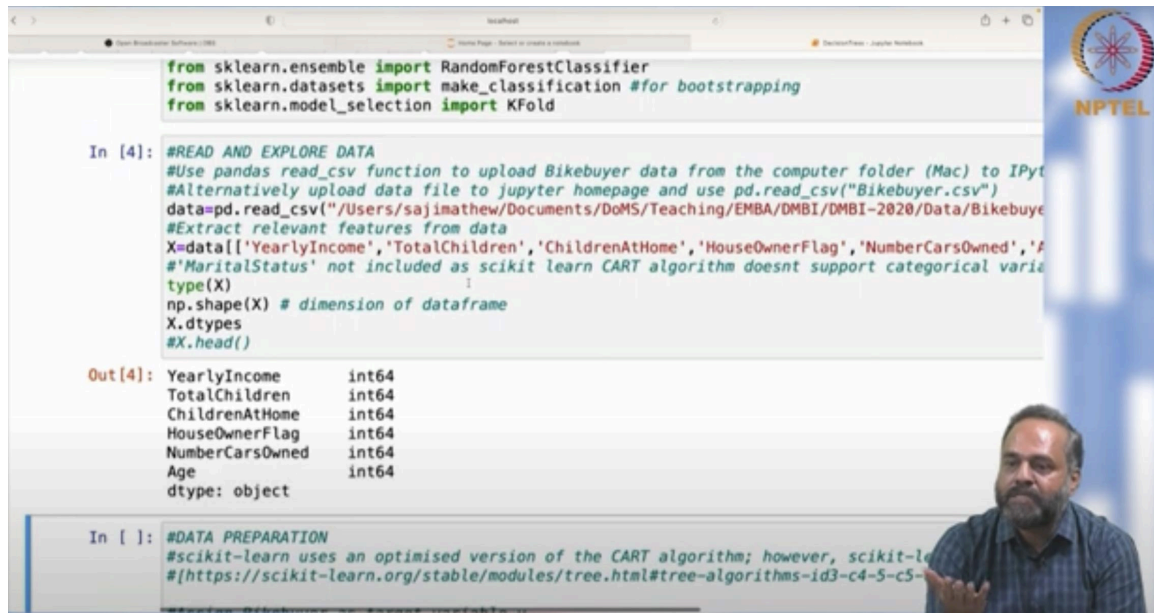
```
from sklearn.metrics import confusion_matrix
from sklearn.tree import export_graphviz
import pandas as pd
import numpy as np
import graphviz
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification #for bootstrapping
from sklearn.model_selection import KFold

In [ ]: #READ AND EXPLORE DATA
#Use pandas read_csv function to upload Bikebuyer data from the computer folder (Mac) to IPyt
#Alternatively upload data file to jupyter homepage and use pd.read_csv("Bikebuyer.csv")
data=pd.read_csv("/Users/sajimathew/Documents/DoMS/Teaching/EMBA/DMBI/DMBI-2020/Data/Bikebuyer
#Extract relevant features from data
X=data[['YearlyIncome', 'TotalChildren', 'ChildrenAtHome', 'HouseOwnerFlag', 'NumberCarsOwned', 'A
#MaritalStatus' not included as scikit learn CART algorithm doesnt support categorical varia
type(X)
np.shape(X), # dimension of dataframe
X.dtypes
X.head()

In [ ]: #DATA PREPARATION
#scikit-learn uses an optimised version of the CART algorithm; however, scikit-learn i
#[https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-
#Assign Bikebuyer as target variable y
y=data['BikeBuyer']
#Split X into train and test data: https://scikit-learn.org/stable/modules/param
```

The file name is bike buyer.csv. So that is the first instruction and you can see that I am using, I am reading it as pd.read csv. Read csv is a function in pandas and it gets read as a pandas data frame, meaning it functions as a multi data type table. So I can access different columns, although their data types are different. And now I need to extract my, I have to separate my data now into target data and x data.

Your predictors and target. So the attributes and the target variables. So x is the attribute data set or the attribute variables. And what does my x consist of? I am going to use, we have seen this labels already. I am going to use yearly income, total children, children at home, house owner flag, number of cars owned and age. These are the variables I am going to use as the x data, as the attributes.



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification #for bootstrapping
from sklearn.model_selection import KFold

In [4]: #READ AND EXPLORE DATA
#Use pandas read_csv function to upload Bikebuyer data from the computer folder (Mac) to IPyt
#Alternatively upload data file to jupyter homepage and use pd.read_csv("Bikebuyer.csv")
data=pd.read_csv("/Users/sajimathew/Documents/DoMS/Teaching/EMBA/DMBI/DMBI-2020/Data/Bikebuye
#Extract relevant features from data
X=data[['YearlyIncome','TotalChildren','ChildrenAtHome','HouseOwnerFlag','NumberCarsOwned'],'#
#MaritalStatus' not included as scikit learn CART algorithm doesnt support categorical varia
type(X)
np.shape(X) # dimension of dataframe
X.dtypes
#X.head()

Out[4]: YearlyIncome      int64
TotalChildren          int64
ChildrenAtHome         int64
HouseOwnerFlag         int64
NumberCarsOwned        int64
Age                    int64
dtype: object

In [ ]: #DATA PREPARATION
#scikit-learn uses an optimised version of the CART algorithm; however, scikit-learn
#[https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-
```

So I am defining that particular data frame. Data is already a data frame in pandas. So that particular data frame is defined as x . x is equal to, from data I am extracting these variables. And if you want to see what is the type of x , you can actually look at here.

I can just execute it here. It is a pandas code data frame. So just to be sure about it. And shape, actually you know you must be familiar shape provides the dimensions of the table. So it is six variables or six columns and how many records? How many records? 18,484 records or rows or tuples. 18,484 tuples, fairly decent size of data and six variables or six attributes.

That is a shape. And what are the data types? The data types of different variables are given here based on the auto identification done on the data. All the data is identified as integers. And now you can see, I have put a comment there marital status not included. I did not include my marital status in my attribute list. And there is a reason for it. And that is a limitation.

So, when I am going to use the CART algorithm. The particular version of the algorithm I am using, it does not support categorical variables. That is a serious limitation. And if I give categorical variables and label them as 1, 2, 3 etc. The algorithm will take it as a

numeric variable. And that will lead to problem. It is actually not a numeric variable. So it will do calculations on categorical data and so on. So in order to avoid that I am just dropping that variable.

I do not want a categorical variable. Others are either interval variables or ordinal variables. We can still use them. But categorical, I am avoiding. But you are smarter than me in coding. Do explore Scikit-learn or any other library, where you can find algorithms that will support all kinds of data.

Here, there is a limitation. And I want to see that in your assignment. And the last one. It will, that, x head dot head is very interesting because it gives you a sense of how the data looks like. So this is our data. It is read into the Jupyter notebook and we are ready to work on this or analyze this data using Scikit-learn functions.

We are all set. We got the data onto the platform. All right. So next step, of course you know, in data mining and analytics data preparation is important step. So we have got the data but we need to look at the data. We need to categorize the data.

And also this is a prepared data. There is not any missing data. There is not any problem with the data set. This is for classroom work and therefore you are getting a data set that is ready for exercise. So in the next cell, I am actually extracting the y data or the target data.

My y data is the bike buyer data. That is my y. So I am extracting it from the data frame data. And now, you remember we called a function from Scikit-learn that is train test split, that function. You can use that function to split the data into train data and test data. And of course, you have to specify arguments.

Your arguments are the x. x is the attributes we already created, y is the target variable and you have to specify what is your test size. And of course, train size already get derived from the test size.

And what does test size equals 0.33 mean? 33 percent, one third of the data will be used for testing or two thirds will be used for training. And random state equals 1 mean, it will be picking one third of the data randomly from the data set. If you make it 0, it won't be random. Now, let's build the model.

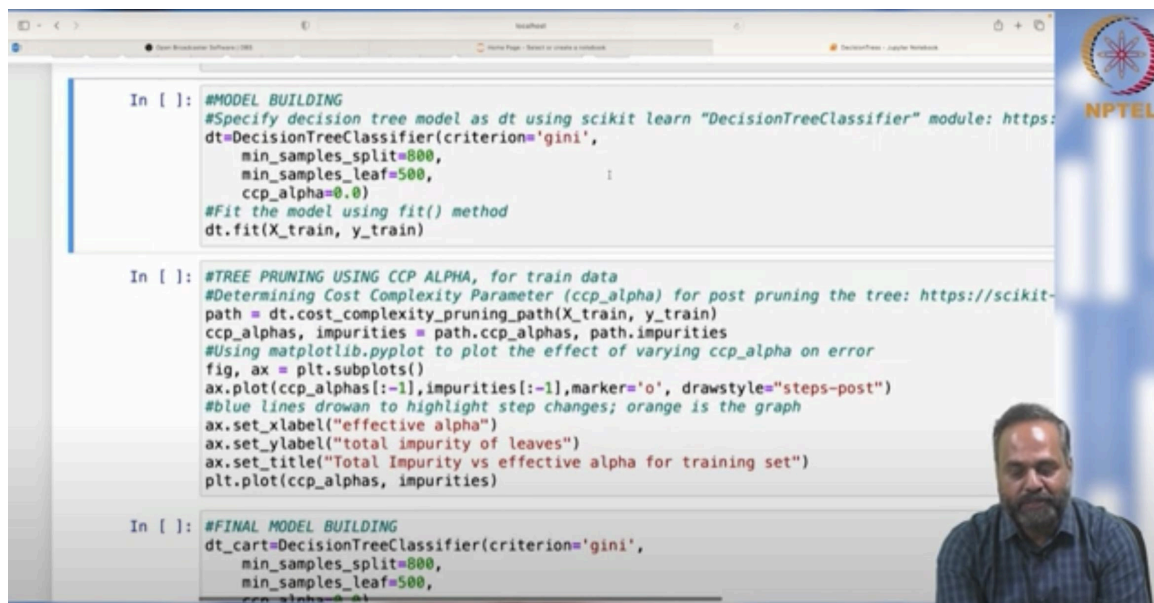
Model building is fairly straightforward. You have to look at the manual, Scikit-learn manual for decision tree class classifier function. It will actually gives a whole description of arguments as well as it also gives you examples. You know, you must be

exploring that. But here, we give certain so called control parameters or hyper parameters.

Parameters that will control the processing of the algorithm. So one is of course, you have to give the method for purity or the measure of purity. And the criterion is that. So I am giving Gini. You remember we discussed what is a Gini score.

You can choose entropy or there are other measures as well. So let us use Gini. And then, there are two hyper parameters or control parameters. Minimum sample split 800 and minimum sample leaf. Unfortunately in different literature and different software, they use different terminologies, that may often confuse you.

Some will call it bucket size. You may wonder what is a bucket size. Bucket size is the minimum leaf. Minimum size of the leaf, I guess in some literature. So 800 is the minimum size required for a node to split. If a node is less than 800, it will not split.



```
In [ ]: #MODEL BUILDING
#Specify decision tree model as dt using scikit learn "DecisionTreeClassifier" module: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
dt=DecisionTreeClassifier(criterion='gini',
    min_samples_split=800,
    min_samples_leaf=500,
    ccp_alpha=0.0)
#Fit the model using fit() method
dt.fit(X_train, y_train)

In [ ]: #TREE PRUNING USING CCP ALPHA, for train data
#Determining Cost Complexity Parameter (ccp_alpha) for post pruning the tree: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.cost_complexity_pruning_path
path = dt.cost_complexity_pruning_path(X_train, y_train)
ccp_alphas, impurities = path.ccp_alphas, path.impurities
#Using matplotlib.pyplot to plot the effect of varying ccp_alpha on error
fig, ax = plt.subplots()
ax.plot(ccp_alphas[:-1], impurities[:-1], marker='o', drawstyle="steps-post")
#blue lines drawn to highlight step changes; orange is the graph
ax.set_xlabel("effective alpha")
ax.set_ylabel("total impurity of leaves")
ax.set_title("Total Impurity vs effective alpha for training set")
plt.plot(ccp_alphas, impurities)

In [ ]: #FINAL MODEL BUILDING
dt_cart=DecisionTreeClassifier(criterion='gini',
    min_samples_split=800,
    min_samples_leaf=500,
    ccp_alpha=0.0)
```

That is that minimum sample to split. And 500 is the minimum size of a leaf or a child node. No child node will be formed which is, or not child node. No leaf node will be formed which is of size lower than 500. This is a specific input we are giving to the algorithm.

So the algorithm has to ensure this is met when it actually runs. And we discussed all this. It may not work. It may not split actually, because when it try to split one of them becomes 200 or 300 etc.

So it has to be minimum 1000. You are right. It may not split. You can check when it actually gets formed. Alright. And CCP α we discussed tree pruning in the class. And here is where we are applying it.

CCP α determines the growth of the tree. And if CCP α is 0, what happens? Will the tree be large or would the tree be shrunk? $\alpha = 0$ means the tree will grow to maximum extent. It is not stopped. It is not pruned. No pruning. That is what CCP $\alpha = 0$ means. And the moment you put a value, it will try to drop leaf nodes and re-evaluate the performance and drop leaves if necessary. So we are starting with 0. We are letting the tree grow to its full extent.

And you can see that dt is the name of the classifier. dt equals. So I am building a tree. I am specifying a tree. This is model specification.

The first step is model specification. Decision tree classifier. All the criterion I give, I am specifying. In the next step, I am fitting. You specified a model does not mean that the model is trained.

In the next step when you fit, the model gets trained. Training has not happened. You are only specifying the model in the previous step. These are subtle things, when you code. Otherwise you get confused what is the difference between fit and applying the function. Decision tree classifier you are applying does not mean that the model is built. Model building happens in the last step when dt is fit into the data, x train and y train. x train and y train. This is the data for x and y for training, we extracted earlier. So this is, this applies to general machine learning algorithmic programming in python. When we use other examples also. I will not have to explain it again but, here we go. Now one of the important steps for us now is to determine what should be the α value. CCP α . Because we just put it 0 but we need an optimum CCP α . So one way to determine that CCP α is to do it empirically. Empirically means you do not derive any value but you try different values, say from 0 to 1 and look at what point the error becomes least. The error becomes least and pick, use that CCP α . And that is what this particular cell is doing.

I have given comments for most of the steps I have used there. So I do not want to get into the details of it. But essentially this code is running or fitting data into the dt which is the model that we have built, for different CCP α . And then check the errors.

Impurities here mean error. They have this interesting term impurity. Impurity means error. And then plot it. Visualize it. Let us see how it is visualized. So, you can see that the x axis is a range of α values and y axis is the corresponding errors or impurities. And

then you can see that when α value is increased from 0 to a higher value, of course it has not gone up till 1. But the trend is that as you increase α , the error is increasing or decreasing? Error is y axis is error.



Impurity is error. It is increasing. It is increasing. Error is increasing. And therefore, it is quite obvious from this that the lowest α is the best for this model or for this problem. If you go to higher α values, the error is, model error is only going to increase. So that is a input for us. So in the next level, what we do? We again specify the model and then we fit the model with the α value 0 because we want to put α 0, which is we already did this.

We do not have to do it again. But that was a trial. But I am finally confirming an α value which is 0.0 ,based on sort of experiment we did.