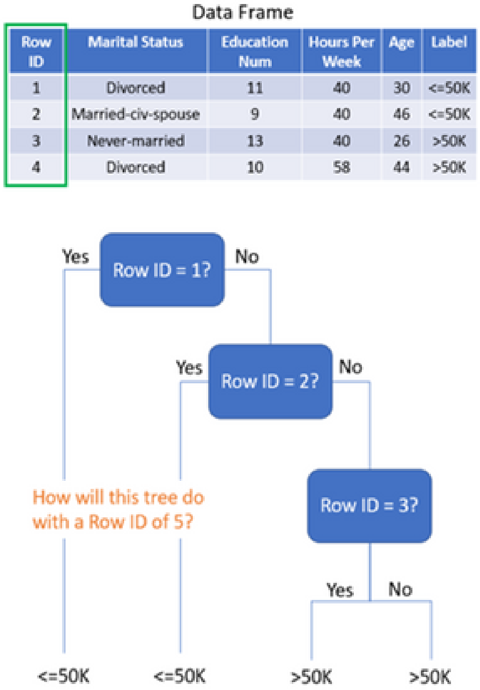
THE MIGHTY RANDOM FOREST ALGORITHM

YOUR MACHINE LEARNING JOURNEY STARTS HERE!

DECISION TREE VARIANCE

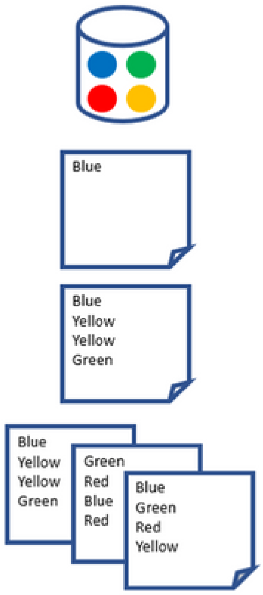


Data Frame

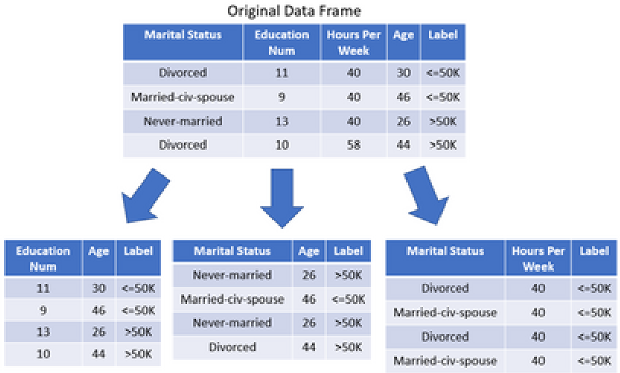
Row ID	Marital Status	Education Num	Hours Per Week	Age	Label
1	Divorced	11	40	30	<=50K
2	Married-civ-spouse	9	40	46	<=50K
3	Never-married	13	40	26	>50K
4	Divorced	10	58	44	>50K

How will this tree do with a Row ID of 5?

BAGGING



FEATURE RANDOMIZATION



Original Data Frame

Marital Status	Education Num	Hours Per Week	Age	Label
Divorced	11	40	30	<=50K
Married-civ-spouse	9	40	46	<=50K
Never-married	13	40	26	>50K
Divorced	10	58	44	>50K

Education Num	Age	Label
11	30	<=50K
9	46	<=50K
13	26	>50K
10	44	>50K

Marital Status	Age	Label
Never-married	26	>50K
Married-civ-spouse	46	<=50K
Never-married	26	>50K
Divorced	44	>50K

Marital Status	Hours Per Week	Label
Divorced	40	<=50K
Married-civ-spouse	40	<=50K
Divorced	40	<=50K
Married-civ-spouse	40	<=50K

Dave ON DATA

Introduction

Interested in machine learning but need help figuring out where to start?

Most applied machine learning (ML) in business uses algorithms based on decision trees. For example, the mighty random forest algorithm. Why?

ML algorithms based on decision trees work very well with tabular data. Think of tables of data in spreadsheets. How common is that in the real world?

It's ubiquitous.

This is a perfect thing for you as you start your ML journey. Not only are ML algorithms based on decision trees state of the art, but they are also the most accessible ML algorithms to learn.

Sound too good to be true?

This document is a brief introduction to the random forest algorithm. Check out the content and then decide.

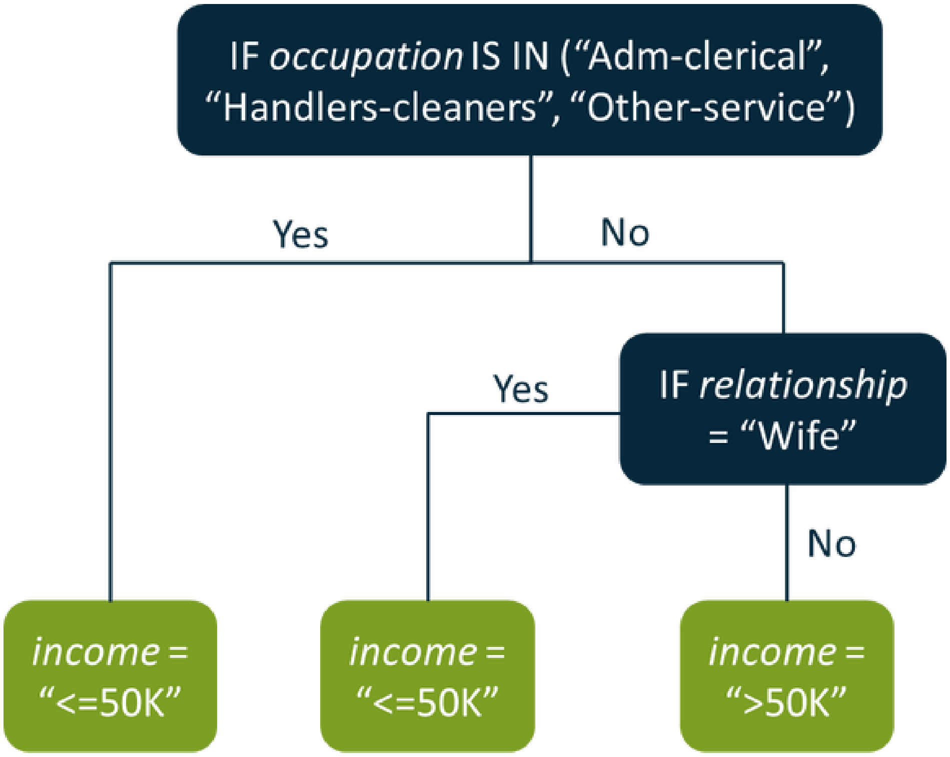
Decision Trees

Almost everyone is familiar with decision trees. Decision trees can take many forms, from flow charts to written rules and graphical representations. The following representations are equivalent:

```
IF occupation IS IN ("Adm-clerical", "Handlers-cleaners", "Other-service")
THEN income = "<=50K"

ELSE IF relationship = "Wife" THEN income = "<=50K"

ELSE income = ">50K"
```



IF *occupation* IS IN ("Adm-clerical", "Handlers-cleaners", "Other-service")

Yes

No

IF *relationship* = "Wife"

Yes

No

income = "<=50K"

income = "<=50K"

income = ">50K"

Wisdom of the Crowd

The random forest algorithm is based on a collection of decision trees working together to make predictions. This arrangement is known as an *ensemble*.

Intuitively, machine learning ensembles rely on the wisdom of the crowd:

"The three conditions for a group to be intelligent are diversity, independence, and decentralization." - James Surowiecki

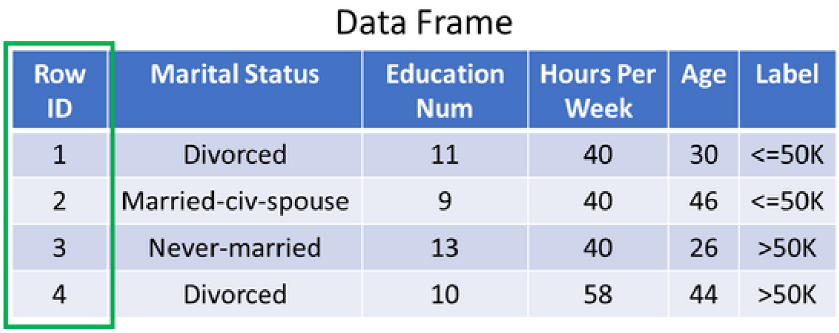
The random forest creates an ensemble of many decision trees. The algorithm ensures that each tree in the forest is independent and diverse, thereby achieving the crowd's wisdom.

The random forest algorithm is democratic. Each tree in the forest has an equal independent vote for making predictions.

Decision Tree Variance

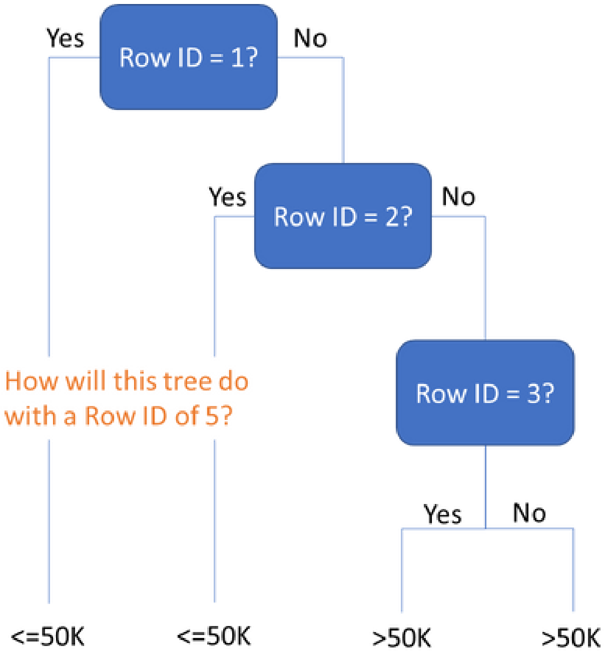
When decision trees are learned from data, their final structure is highly dependent on the data used. Change the data, and you can get trees that look radically different from one another.

In machine learning, this is known as having high *variance*. The example below illustrates how adding the *Row ID* column changes the results.



Data Frame

Row ID	Marital Status	Education Num	Hours Per Week	Age	Label
1	Divorced	11	40	30	<=50K
2	Married-civ-spouse	9	40	46	<=50K
3	Never-married	13	40	26	>50K
4	Divorced	10	58	44	>50K



How will this tree do with a Row ID of 5?

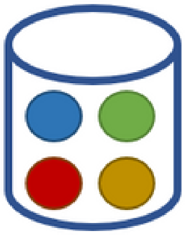
The random forest algorithm leverages decision tree variance to manufacture diversity across all the trees in the forest.

Bagging

We know that individual decision trees will change their shape depending on the data provided for training (i.e., decision trees have high variance).

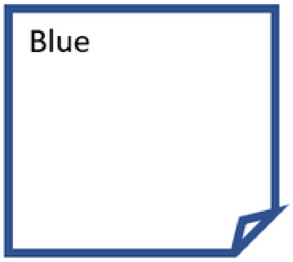
Bagging (aka “bootstrap aggregation”) is a technique that allows for taking a single data set and “manufacturing” many different data sets. Bagging works via *random sampling with replacement*:

I have a jar with 4 marbles – blue, green, red, and yellow.



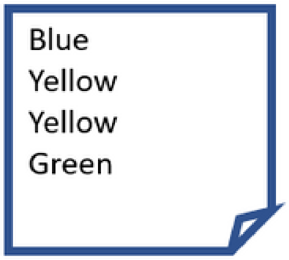
I reach into the jar and draw a marble at random.

I write down the marble I drew and put it back.



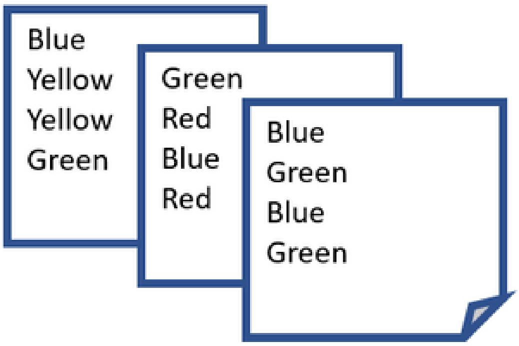
Blue

I repeat this process three times, resulting in a list of marbles that differs from the marbles in the jar.



Blue
Yellow
Yellow
Green

I can make as many lists as I would like.



Blue
Yellow
Yellow
Green

Green
Red
Blue
Red

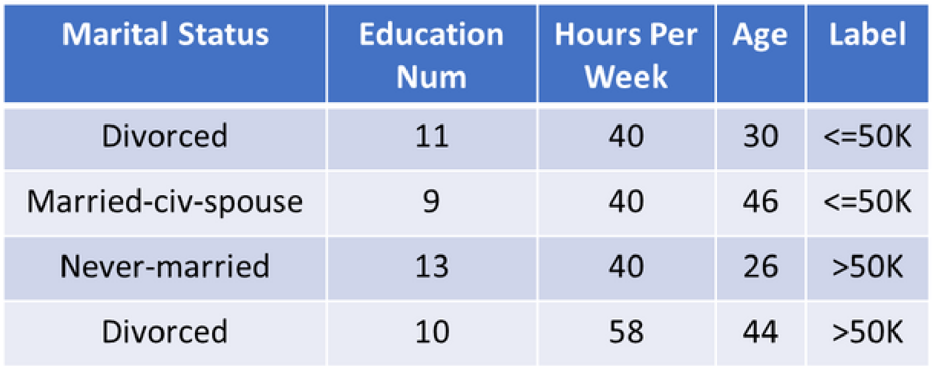
Blue
Green
Blue
Green

Bagging

The mighty random forest uses bagging to create different data sets to train the trees in the forest. Because the trees have different data, they will be different.

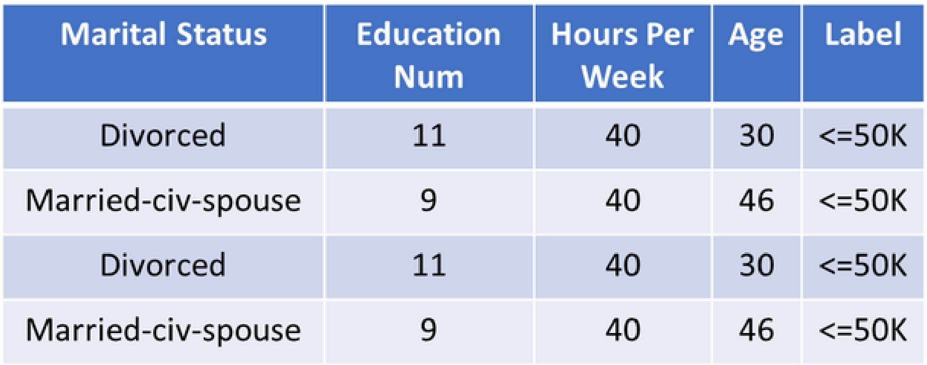
Per the “wisdom of the crowd,” we want the trees to be wildly different from each other. Bagging helps to achieve that goal.

Original Data Frame



Marital Status	Education Num	Hours Per Week	Age	Label
Divorced	11	40	30	<=50K
Married-civ-spouse	9	40	46	<=50K
Never-married	13	40	26	>50K
Divorced	10	58	44	>50K

“Manufactured” Data Frame



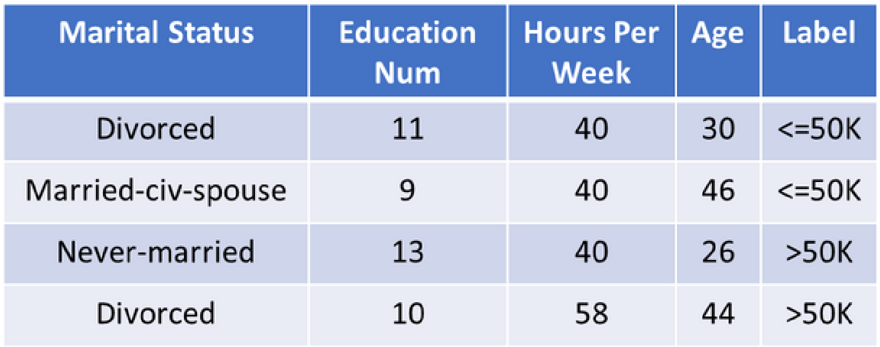
Marital Status	Education Num	Hours Per Week	Age	Label
Divorced	11	40	30	<=50K
Married-civ-spouse	9	40	46	<=50K
Divorced	11	40	30	<=50K
Married-civ-spouse	9	40	46	<=50K

Feature Randomization

Bagging allows us to randomize the rows (i.e., observations) of data used to train each tree in the forest. This randomization makes the trees different from each other.

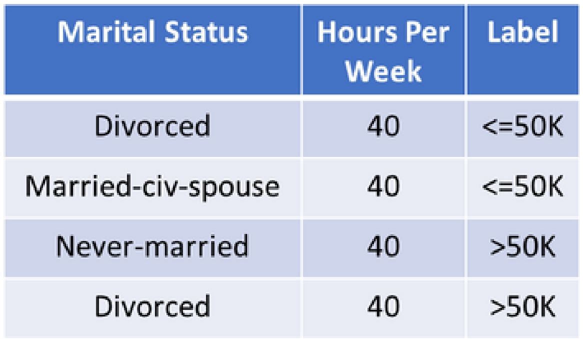
Our goal, however, is for the trees to be wildly different from each other. What if we randomize the columns (i.e., features) to a subset as well?

Original Data Frame



Marital Status	Education Num	Hours Per Week	Age	Label
Divorced	11	40	30	<=50K
Married-civ-spouse	9	40	46	<=50K
Never-married	13	40	26	>50K
Divorced	10	58	44	>50K

“Manufactured” Data Frame

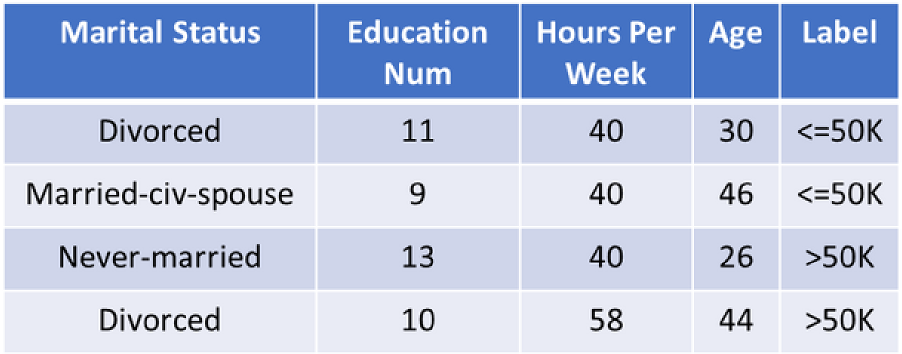


Marital Status	Hours Per Week	Label
Divorced	40	<=50K
Married-civ-spouse	40	<=50K
Never-married	40	>50K
Divorced	40	>50K

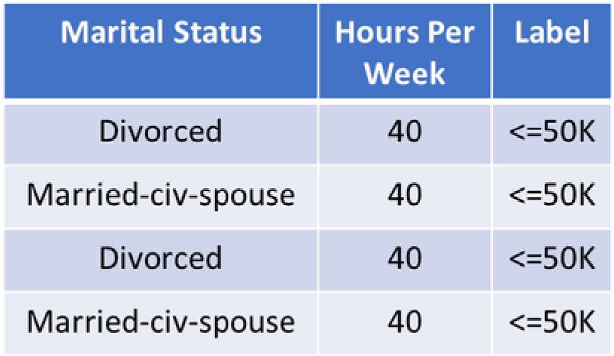
THE MIGHTY RANDOM FOREST

The secret sauce of the mighty random forest is that it builds many wildly different trees by randomizing both the rows and columns:

Original Data Frame

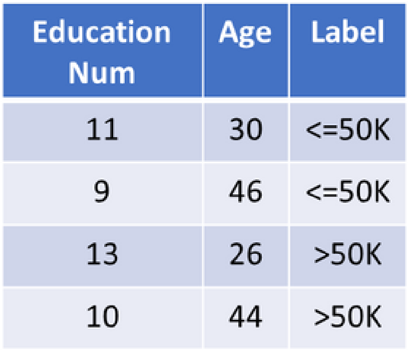


Marital Status	Education Num	Hours Per Week	Age	Label
Divorced	11	40	30	<=50K
Married-civ-spouse	9	40	46	<=50K
Never-married	13	40	26	>50K
Divorced	10	58	44	>50K



Marital Status	Hours Per Week	Label
Divorced	40	<=50K
Married-civ-spouse	40	<=50K
Divorced	40	<=50K
Married-civ-spouse	40	<=50K

"Manufactured" Data Frame 1



Education Num	Age	Label
11	30	<=50K
9	46	<=50K
13	26	>50K
10	44	>50K

"Manufactured" Data Frame 2

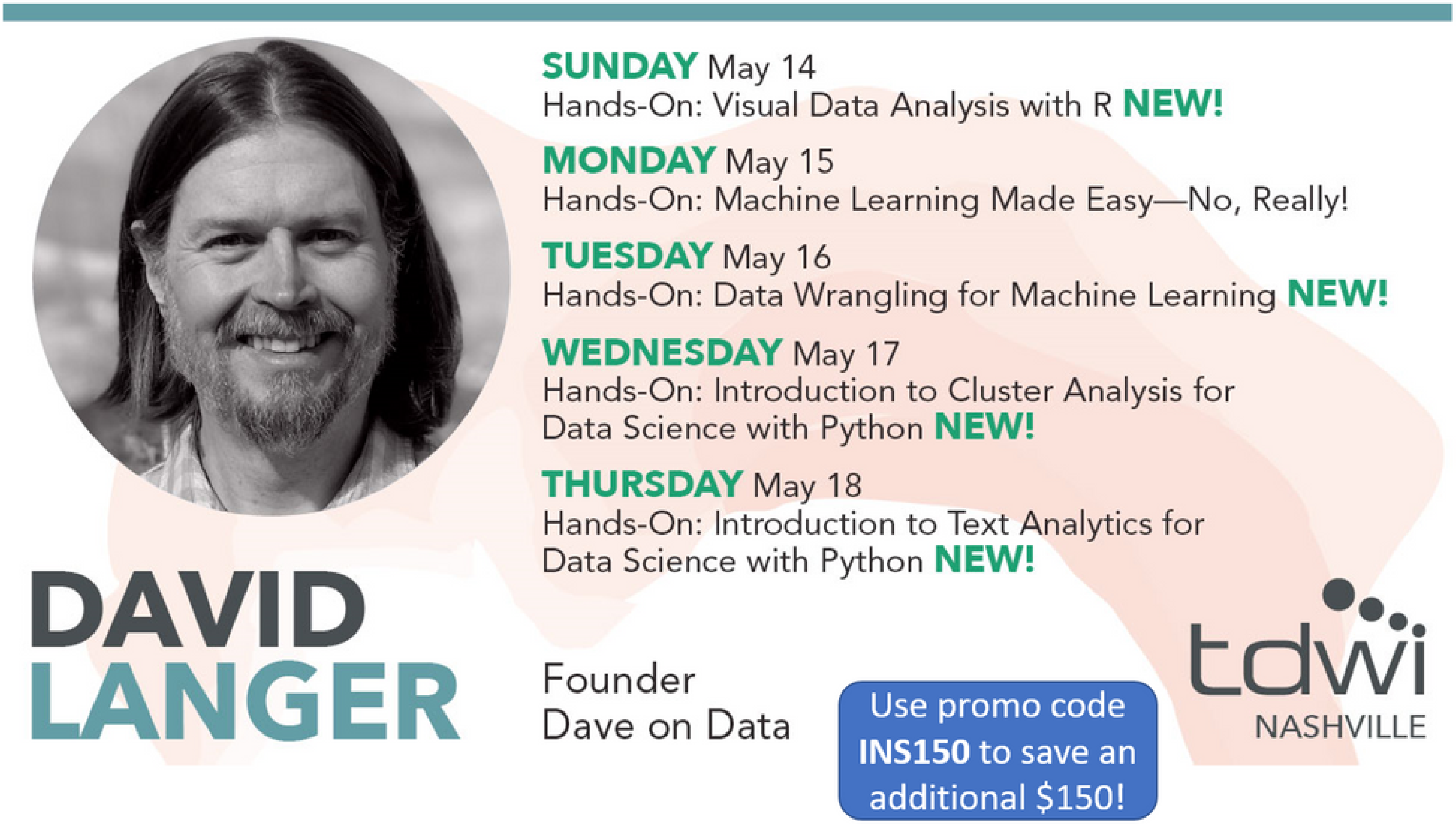
Using the manufactured data, Tree 1 will be wildly different from Tree 2!

Want to Learn More?

The content in this document comes from the following live training course:

- Machine Learning Made Easy - No, Really!

My hands-on live courses are consistently top-rated by conference attendees. If you learn best via a hands-on, in-person classroom experience, check out the link below.



SUNDAY May 14
Hands-On: Visual Data Analysis with R **NEW!**

MONDAY May 15
Hands-On: Machine Learning Made Easy—No, Really!

TUESDAY May 16
Hands-On: Data Wrangling for Machine Learning **NEW!**

WEDNESDAY May 17
Hands-On: Introduction to Cluster Analysis for Data Science with Python **NEW!**

THURSDAY May 18
Hands-On: Introduction to Text Analytics for Data Science with Python **NEW!**

DAVID LANGER

Founder
Dave on Data

Use promo code **INS150** to save an additional $150!

tdwi
NASHVILLE

https://bit.ly/TDWINashville2023

About the Author

My name is Dave Langer and I am the founder of Dave on Data.

I'm a hands-on analytics professional, having used my skills with Excel, SQL, and R/Python to craft insights, advise leaders, and shape company strategy.

I'm also a skilled educator, having trained 100s of working professionals in live in-person classroom settings and 1000s more via live virtual training and online courses.

In the past, I've held analytics leaderships roles at Schedulicity, Data Science Dojo, and Microsoft.

Drop me an email if you have any questions: dave@daveondata.com