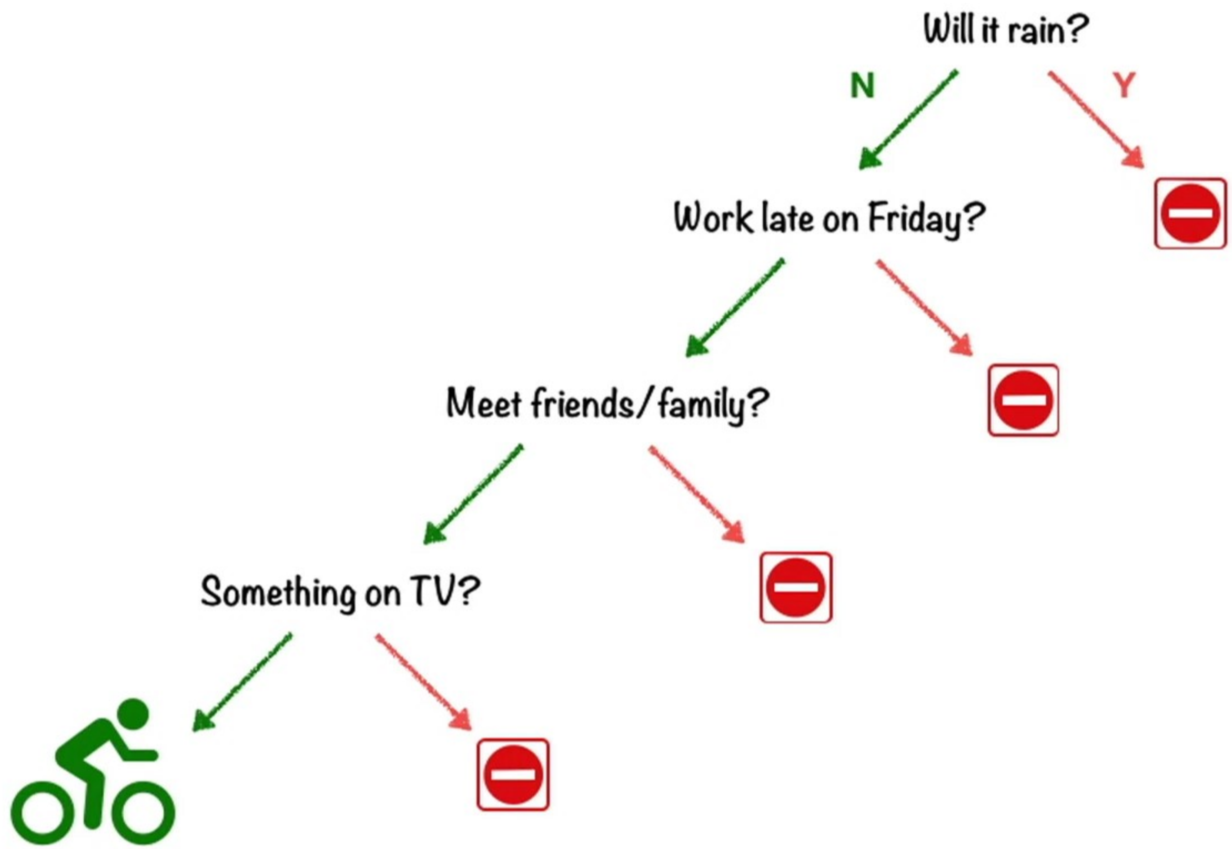
Decision Trees

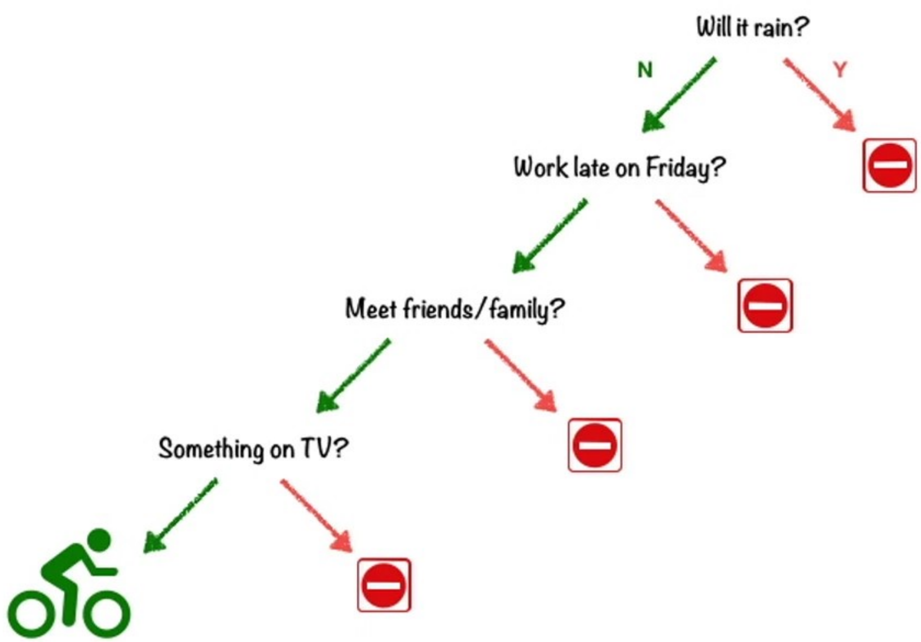
Should I go biking on Saturday morning?

- Will it rain?
- Will I work late on Friday?
- Do I need to meet friends/family?
- Is there something on TV?

Should I go biking on Saturday morning?



Decision Tree



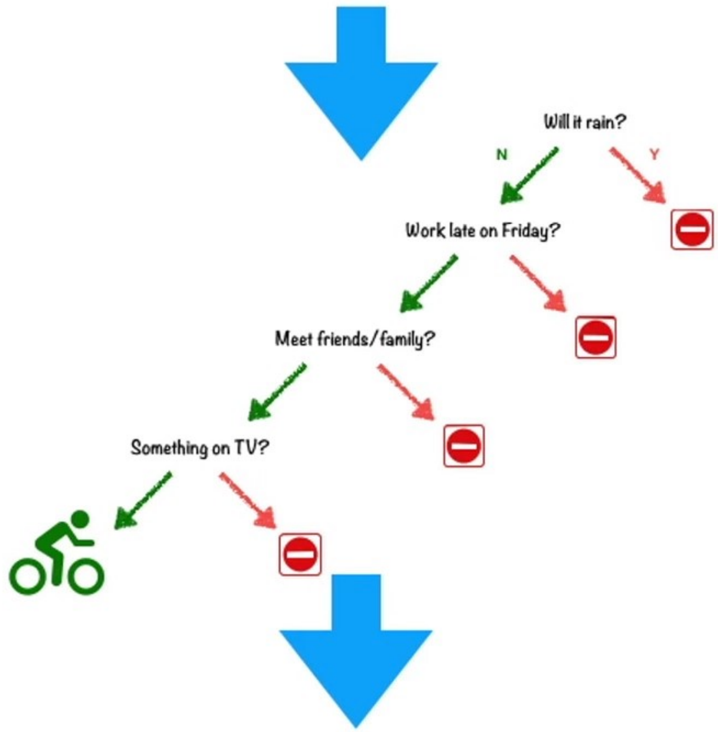
This is exactly what a decision tree looks like

Decision Tree

- Helps predict an outcome given a set of inputs
- In business, it represents visually how a decision is taken with inputs and consequences of each decision

Decision Tree

Input Variables/Predictors



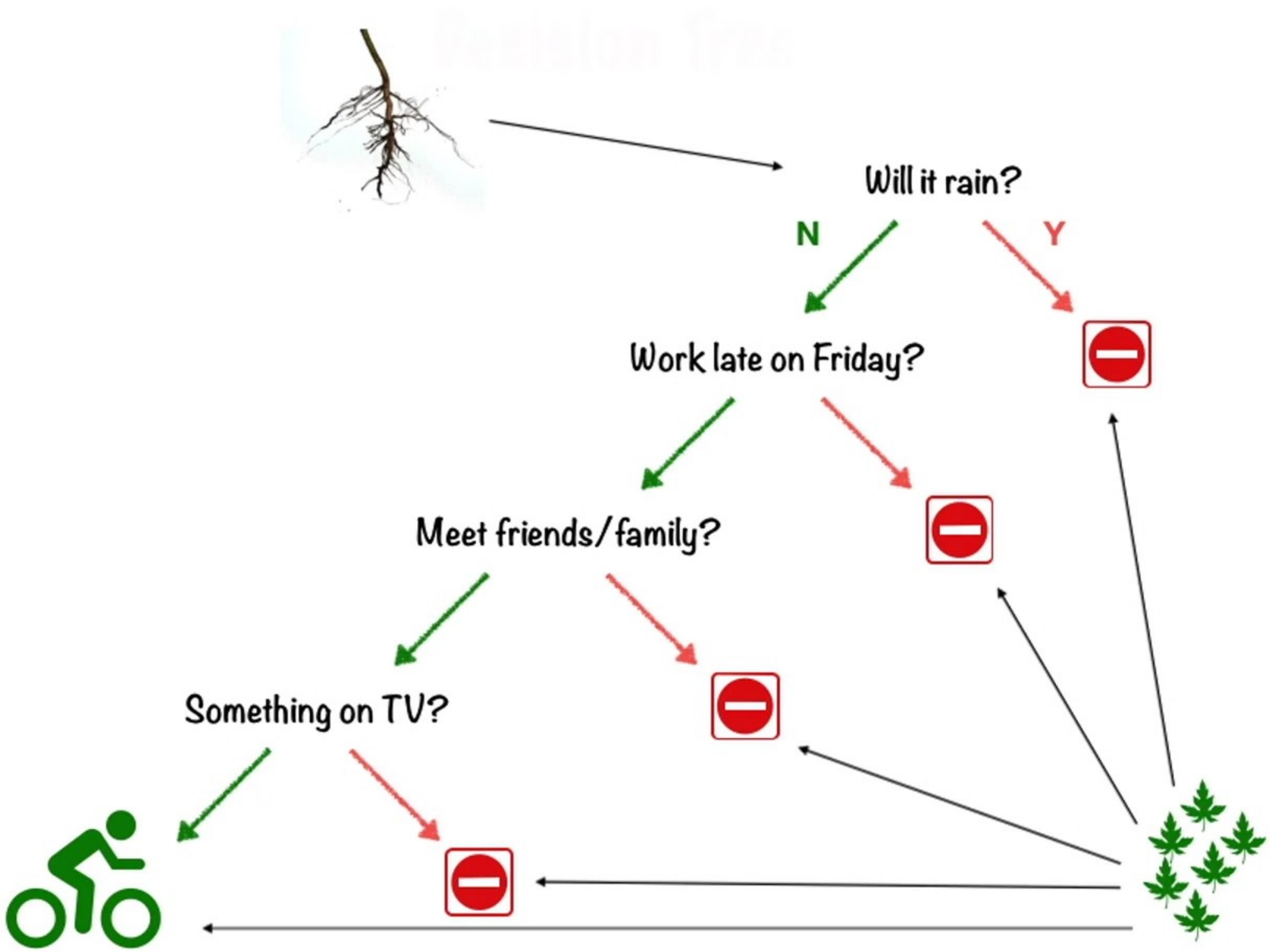
Outcome/Output Variables

Decision Tree

- Can be used for classification, just like SVM
- With SVM, can't understand relationship between input variables and outcome
- Decision Trees are not a blackbox

Decision Tree

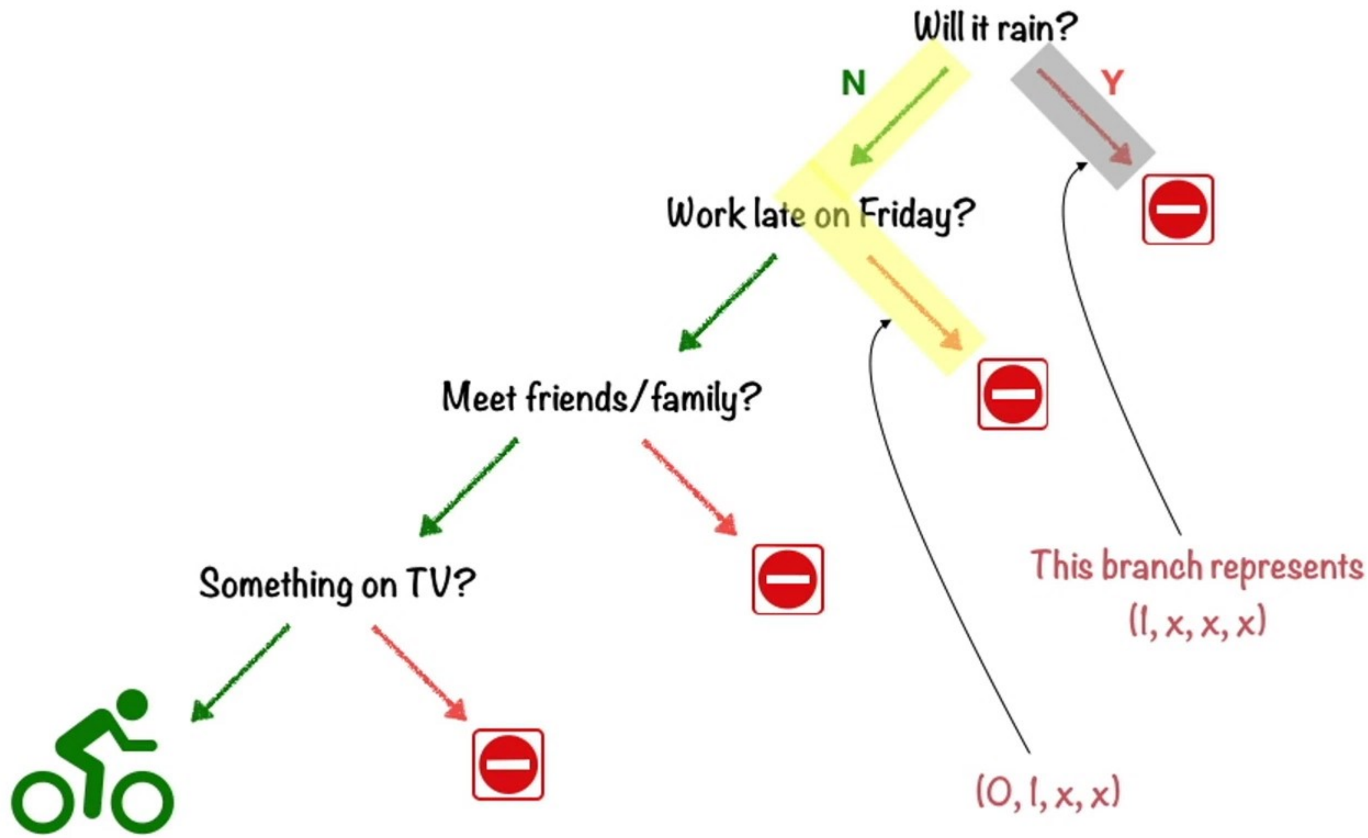
- Inputs can be categorical or continuous
- Outputs can also be categorical or continuous
  - Called regression tree when outcome is continuous


Will it rain?
N
Y
Work late on Friday?
Meet friends/family?
Something on TV?

Decision Tree

- The leaves of a tree are the outcomes
- In a classification problem, these are class labels
- The tree gives the most likely outcome given the values of the variables
- The branches are combinations of predictor values leading to an outcome

Decision Tree



Decision Tree Learning

- The process of creating/learning a decision tree from training data
- Start with training data in the form of feature vector, label/outcome
  - (1,0, 1, 1), No biking
  - (0,0,0,0), I bike!
- Obtain a decision tree used to classify/predict a new instance
- A supervised learning approach

Decision Tree Learning

- Recursive partitioning is the most common strategy for decision tree learning
- Decision tree learning algorithms
  - CART
  - ID3
  - C4.5
  - CHAID

Key takeaways from this chapter

What is Decision Tree?
A decision tree is a graphical representation of all the possible solutions to a decision based on certain conditions. It's called a decision tree because it starts with a single box (or root), which then branches off into a number of solutions, just like a tree.

We will start with the training data in the form of feature vector, label/outcome.

Types of Decision Tree Algorithms
CART
ID3
C4.5
CHAID

Decision Tree Learning

- Algorithm needs to tell us the order in which the predictors are evaluated
- If the predictor is continuous, the tree needs to split the variable into ranges

Greedy Algorithm for Learning a Decision Tree

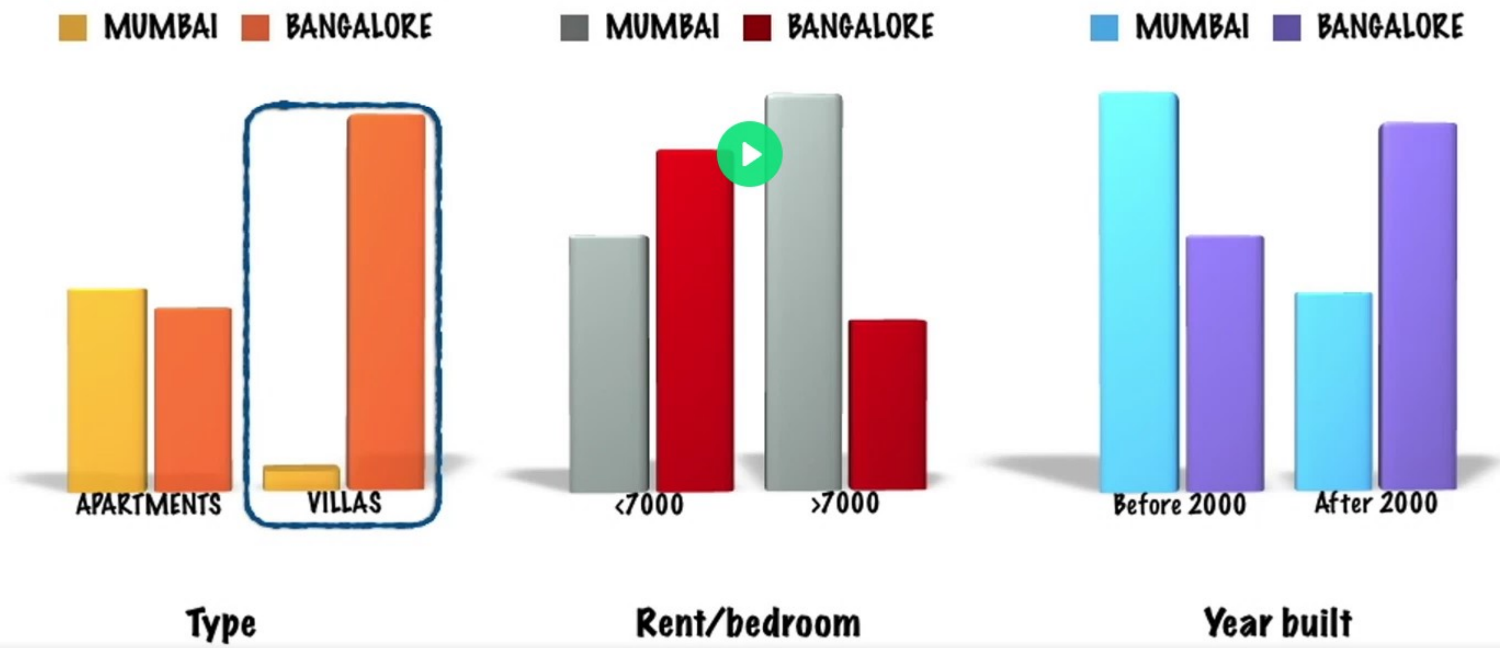
GIVEN THE
TYPE OF HOUSING,
RENT/BEDROOM AND
THE YEAR IT WAS BUILT

PREDICT THE CITY TO
WHICH A RESIDENCE
BELONGS

MUMBAI (OR)
BANGALORE

Greedy Algorithm for Learning a Decision Tree

DRAW A HISTOGRAM FOR EACH ATTRIBUTE, FOR RESIDENCES IN EACH CITY

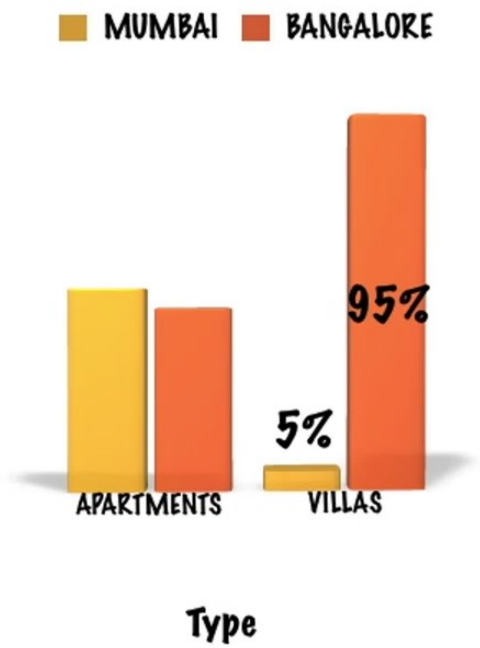


-05:49

THE TYPE OF HOUSING SEEMS TO BE THE CLEAREST INDICATOR OF WHETHER A RESIDENCE BELONGS TO MUMBAI OR BANGALORE

IF IT IS A VILLA, THEN IT MOST LIKELY BELONGS TO BANGALORE

IF IT IS AN APARTMENT, WE ARE STILL NOT SURE WHICH CITY IT BELONGS TO



WE NOW HAVE THE FIRST
NODE OF OUR DECISION TREE

WHAT TYPE OF
RESIDENCE?

APARTMENT

VILLA

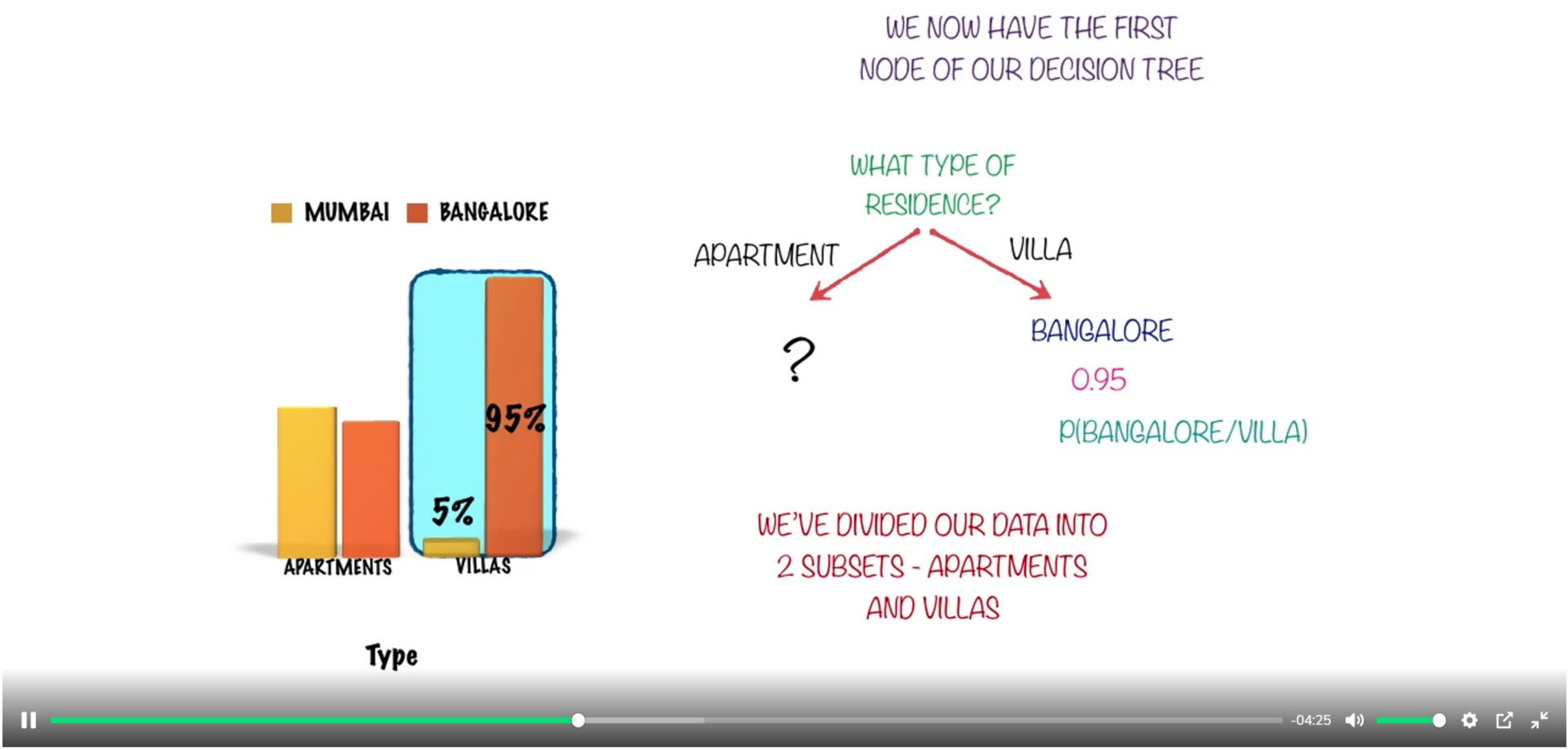
?

BANGALORE

0.95

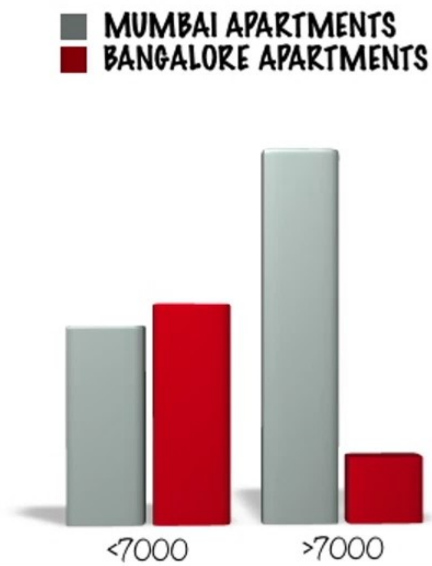
P(BANGALORE/VILLA)

WE'VE DIVIDED OUR DATA INTO
2 SUBSETS - APARTMENTS
AND VILLAS


WE NOW HAVE THE FIRST
NODE OF OUR DECISION TREE
MUMBAI
BANGALORE
95%
5%
APARTMENTS
VILLAS
Type
WHAT TYPE OF
RESIDENCE?
APARTMENT
VILLA
?
BANGALORE
0.95
P(BANGALORE/VILLA)
WE'VE DIVIDED OUR DATA INTO
2 SUBSETS - APARTMENTS
AND VILLAS
-04:25

REPEAT THIS PROCESS RECURSIVELY, FOR EACH SUBSET

DRAW A HISTOGRAM FOR EACH OF THE REMAINING ATTRIBUTES FOR ONLY THE APARTMENTS IN EACH CITY

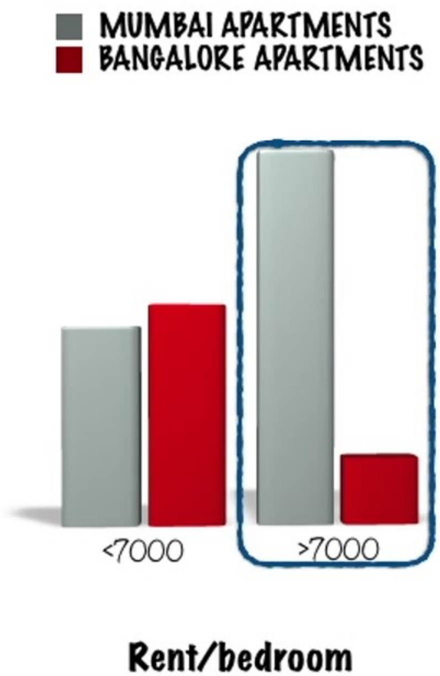


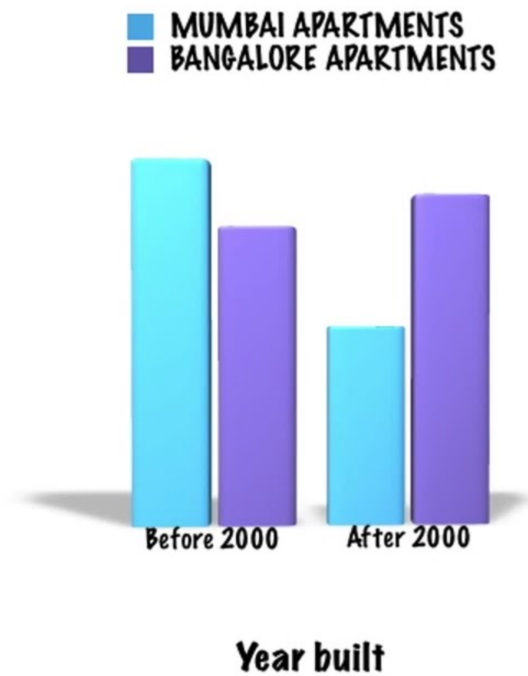
Rent/bedroom

Year built

REPEAT THIS PROCESS RECURSIVELY, FOR EACH SUBSET

DRAW A HISTOGRAM FOR EACH OF THE REMAINING ATTRIBUTES FOR ONLY THE APARTMENTS IN EACH CITY

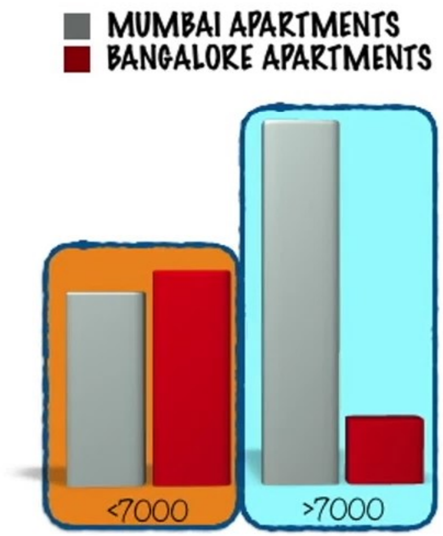




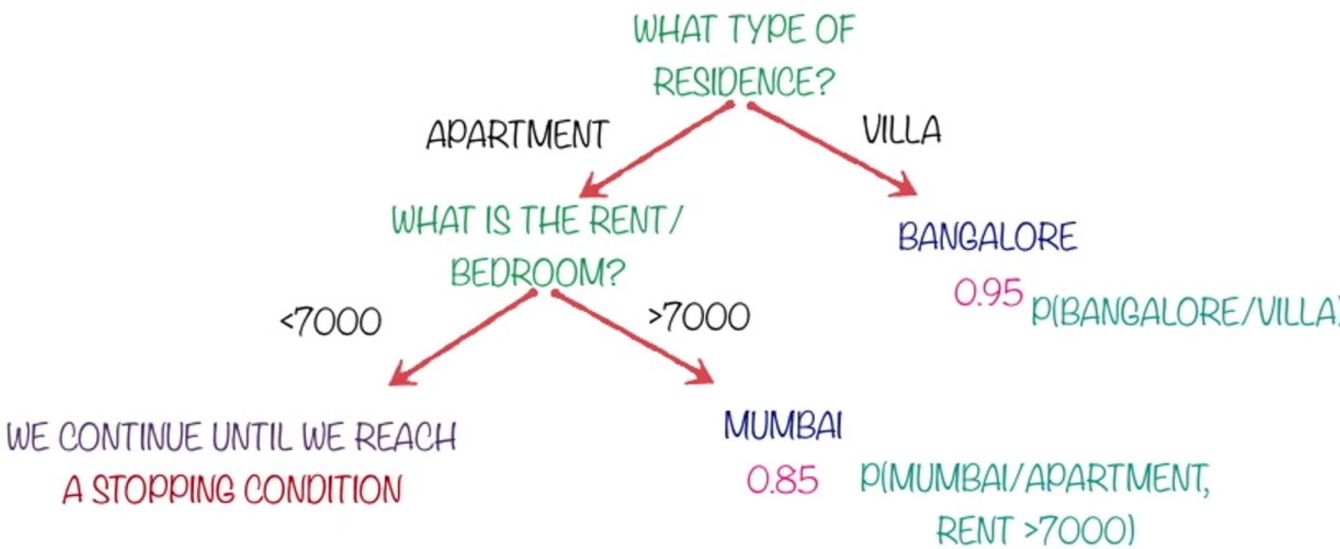

MUMBAI APARTMENTS
BANGALORE APARTMENTS
<7000
>7000
-03:39

WHEN YOU LOOK ONLY AT APARTMENTS, THE RENT/ BEDROOM SEEMS TO BE A GOOD PREDICTOR OF THE CITY

NOW, WE'LL DIVIDE THE APARTMENTS INTO TWO SUBSETS, RENT >7000, RENT <7000

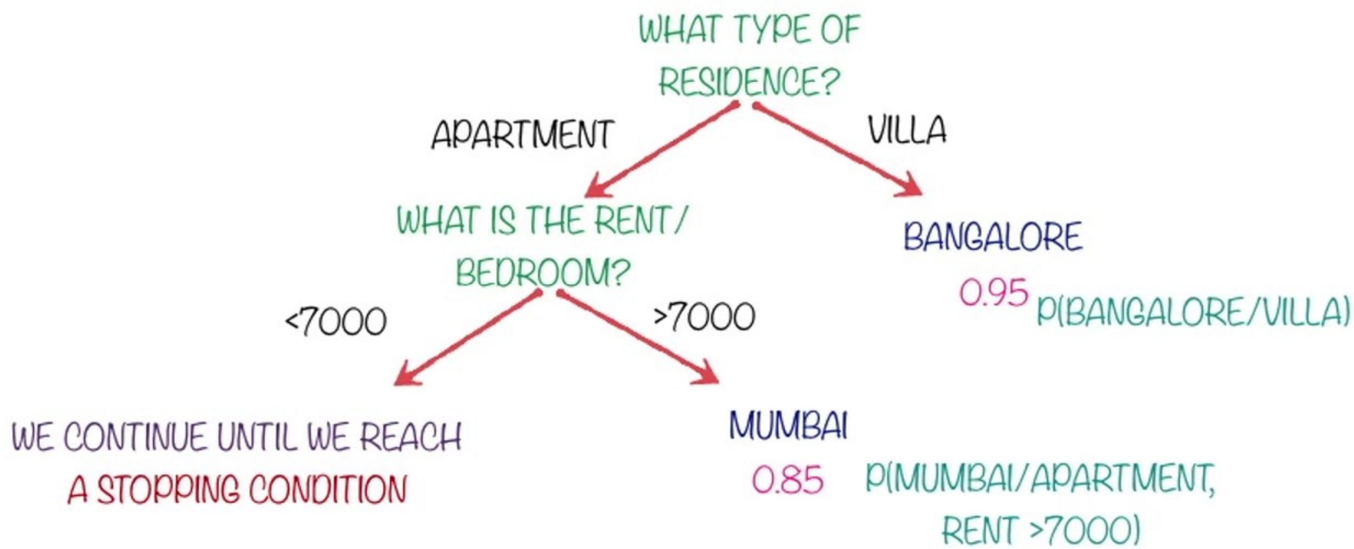


Our Decision Tree so far



This is Recursive Partitioning

Our Decision Tree so far

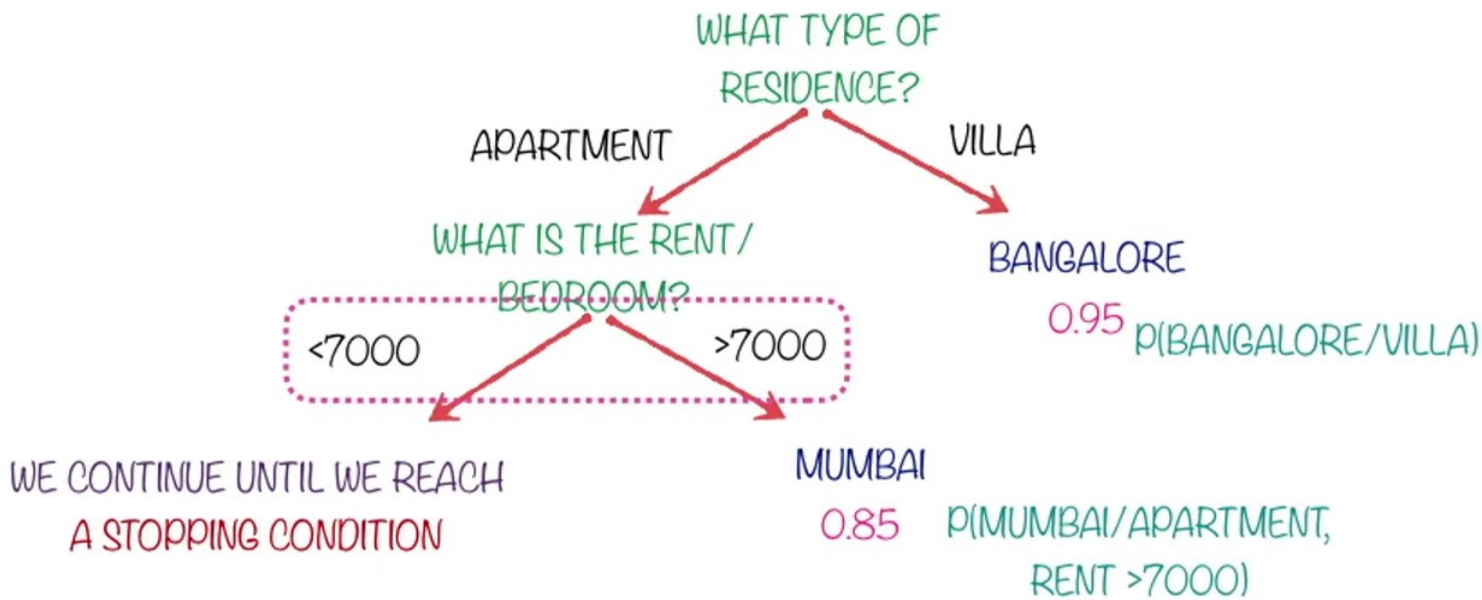


This is Recursive Partitioning

The Stopping Condition could be

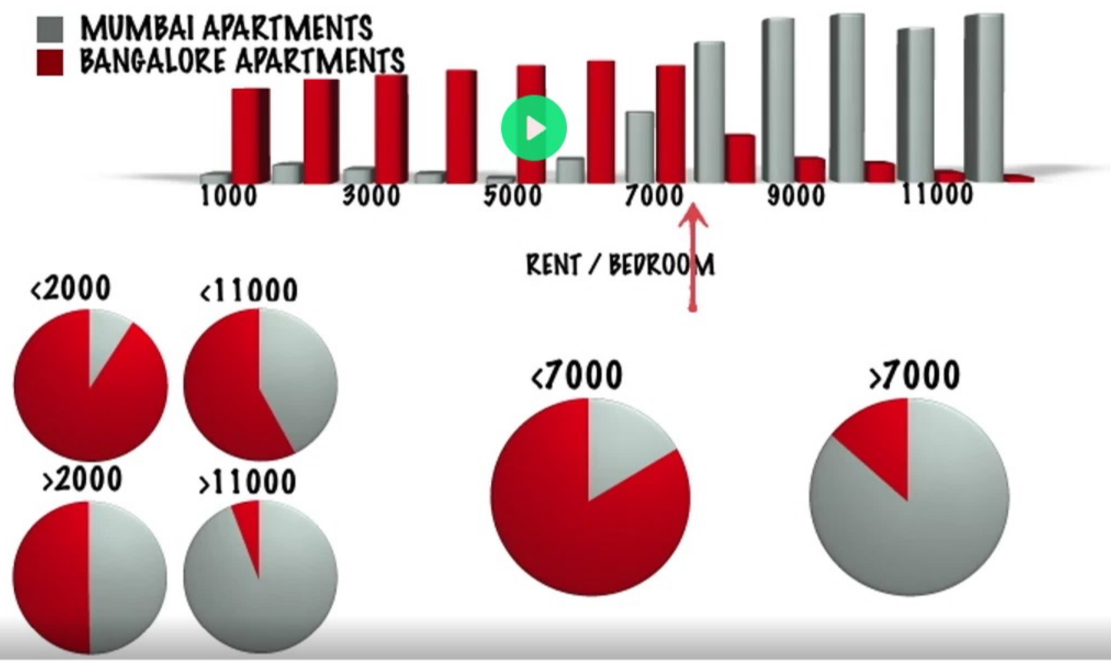
- All our subsets are mostly homogeneous
- We have run out of attributes
- The tree is too large

The Best Split for a Continuous Input Variable



The Best Split for a Continuous Input Variable

The point where the subsets we get are mostly homogeneous



Key takeaways from this chapter

Building the Decision Tree : Decision Tree Learning
Consider the Dataset of Housing Units in a city with Rent, Type of Housing Unit and the city it is in, The year it was built

Recursive partitioning creates a decision tree that strives to correctly classify members of the population by splitting it into sub-populations based on several contratry independent variables.
-All our subsets are mostly homogeneous
-We have run out of attributes
-The tree is too large
The Stopping Condition could be

Choosing the best split for Continuous Input Variable is at the point where the subsets we get are mostly homogeneous.
Take the Housing Rent for Example we got that under and Above Rs.7000 we usually have homogeneous split between the cities.

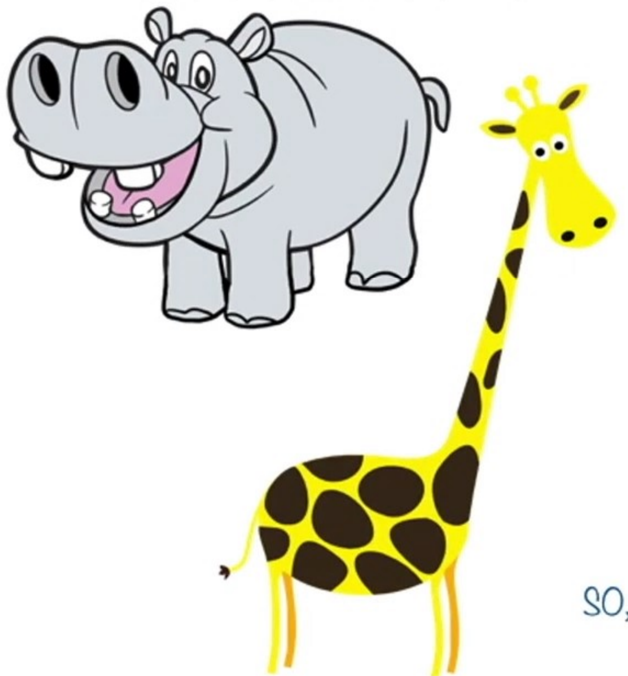
Information Gain

ANY STATEMENT , NEWS OR MESSAGE
CONTAINS INFORMATION

SOME HAVE MORE INFORMATION AND
SOME LESS

THE IDEA OF INFORMATION GAIN IS TO REDUCE
ENTROPY AND MAXIMIZE INFORMATION

LET'S SAY YOU HAVE TO CLASSIFY AN ANIMAL
AS A GIRAFFE OR A HIPPO



IF YOU WERE TOLD, THIS ANIMAL HAS 4 LEGS

THIS IS BASICALLY USELESS! BOTH GIRAFFES AND HIPPOS HAVE 4 LEGS, SO THIS STATEMENT GIVES US NO INFORMATION

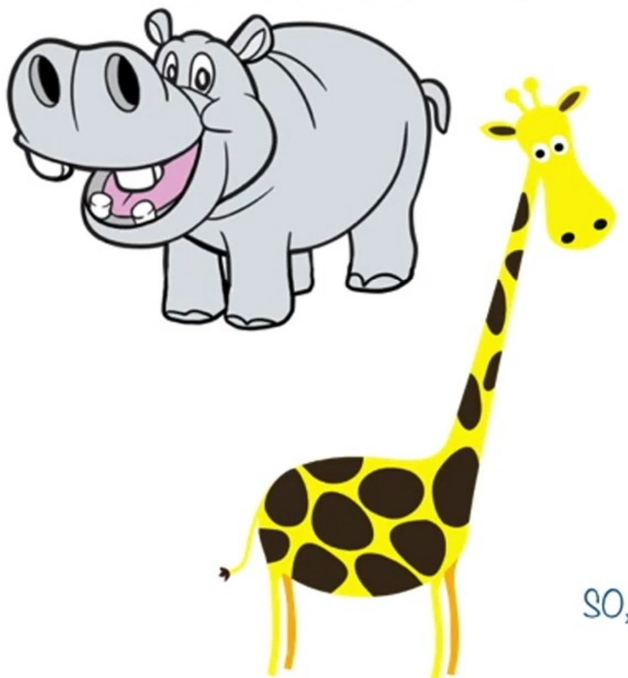
BUT IF YOU WERE TOLD, THIS ANIMAL IS 10 FEET TALL

THIS IS USEFUL INFORMATION!

IT TELLS YOU THAT THE ANIMAL IS VERY LIKELY A GIRAFFE

SO, CLEARLY - THE VALUES OF SOME ATTRIBUTES GIVE US MORE INFORMATION THAN OTHERS

LET'S SAY YOU HAVE TO CLASSIFY AN ANIMAL
AS A GIRAFFE OR A HIPPO



IF YOU WERE TOLD, THIS ANIMAL HAS 4 LEGS

THIS IS BASICALLY USELESS! BOTH GIRAFFES AND HIPPOS HAVE 4 LEGS, SO THIS STATEMENT GIVES US NO INFORMATION

BUT IF YOU WERE TOLD, THIS ANIMAL IS 10 FEET TALL

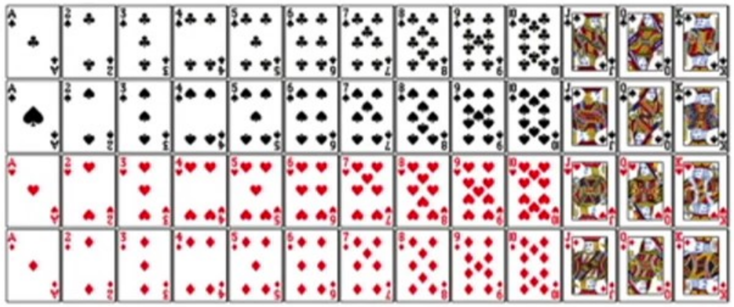
THIS IS USEFUL INFORMATION!

IT TELLS YOU THAT THE ANIMAL IS
VERY LIKELY A GIRAFFE

SO, CLEARLY - THE VALUES OF SOME ATTRIBUTES GIVE US MORE INFORMATION THAN OTHERS

AND THERE IS A MATHEMATICAL WAY TO MEASURE
THIS INFORMATION

INITIALLY, THERE ARE 52 POSSIBLE OUTCOMES IN ALL



YOU ASK

IS THE CARD AN ACE?

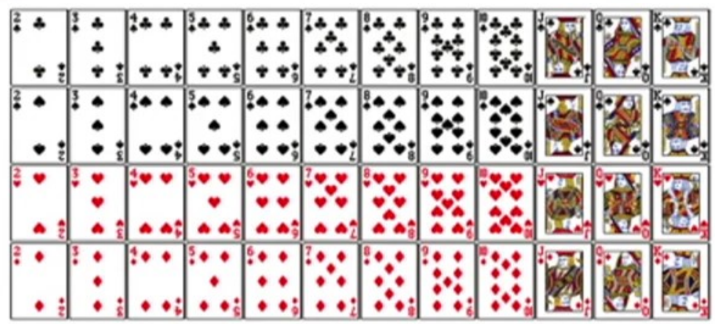
LEFT WITH 48 POSSIBLE OUTCOMES

YES

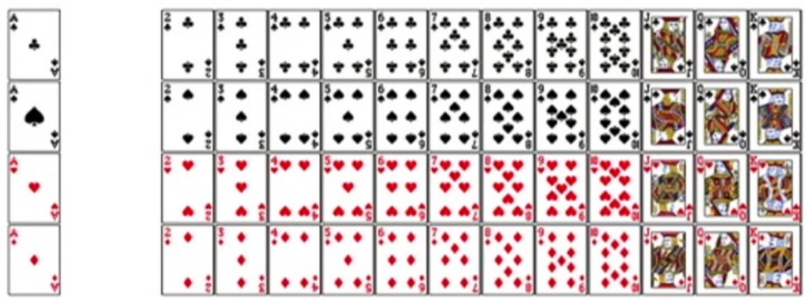
NO

LEFT WITH 4 POSSIBLE OUTCOMES

THE ANSWER "YES" GIVES US MORE INFORMATION THAN THE ANSWER "NO"



YES NO



P(YES)=4/52

P(NO)=48/52

IS THE CARD AN ACE?

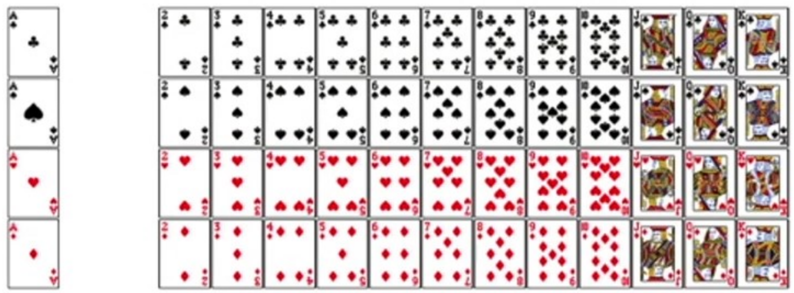
THE ANSWER “YES” GIVES US MORE INFORMATION THAN THE ANSWER “NO”

THE ANSWER “YES” HAS A LOWER PROBABILITY

THE LOWER THE PROBABILITY OF THE ANSWER, THE MORE INFORMATION YOU GET

IF X IS A RANDOM VARIABLE THAT REPRESENTS THE ANSWER TO OUR QUESTION

YES NO



P(YES)=4/52

P(NO)=48/52

IS THE CARD AN ACE?

THE ANSWER “YES” GIVES US MORE INFORMATION THAN THE ANSWER “NO”

THE ANSWER “YES” HAS A LOWER PROBABILITY

THE LOWER THE PROBABILITY OF THE ANSWER, THE MORE INFORMATION YOU GET

IF X IS A RANDOM VARIABLE THAT REPRESENTS THE ANSWER TO OUR QUESTION

INFORMATION CONTENT OF (X=YES) = -LOG(P(X=YES))

INFORMATION CONTENT OF (X=NO) = -LOG(P(X=NO))

INFORMATION CONTENT OF (X=x) = -LOG(P(X=x))

Information Gain

ANY STATEMENT , NEWS OR MESSAGE
CONTAINS INFORMATION

SOME HAVE MORE INFORMATION AND
SOME LESS

THE IDEA OF INFORMATION GAIN IS TO REDUCE
ENTROPY AND MAXIMIZE INFORMATION

IF X IS A RANDOM VARIABLE THAT REPRESENTS THE ANSWER TO OUR QUESTION

INFORMATION CONTENT OF $(X=x) = -\log(P(X=x))$

AVERAGE VALUE OF THE INFORMATION CONTENT (ALSO CALLED THE EXPECTED VALUE)=

$$\boxed{\sum P(X=x)\,(-\log(P(X=x)))}$$

ENTROPY $H(X)$

ENTROPY IS THE AMOUNT OF UNCERTAINTY/UNPREDICTABILITY THERE IS IN THE ANSWER

ENTROPY INCREASES WITH

1) NUMBER OF POSSIBLE ANSWERS
2) THE EVENNESS OF THE PROBABILITY DISTRIBUTION

P(YES) = 0 => THERE IS NO UNCERTAINTY => ENTROPY =0

YES AND NO HAVE EQUAL PROBABILITY => VERY HIGH ENTROPY

IF X IS A RANDOM VARIABLE THAT REPRESENTS THE ANSWER TO OUR QUESTION

INFORMATION CONTENT OF $(X=x) = -\log(P(X=x))$

AVERAGE VALUE OF THE INFORMATION CONTENT (ALSO CALLED THE EXPECTED VALUE)=

$$\boxed{\sum P(X=x)\,(-\log(P(X=x))}$$

ENTROPY $H(X)$

ENTROPY IS THE AMOUNT OF UNCERTAINTY/UNPREDICTABILITY THERE IS IN THE ANSWER

ENTROPY INCREASES WITH

1) NUMBER OF POSSIBLE ANSWERS
2) THE EVENNESS OF THE PROBABILITY DISTRIBUTION

$P(\text{YES}) = 0$ => THERE IS NO UNCERTAINTY => ENTROPY =0

YES AND NO HAVE EQUAL PROBABILITY => VERY HIGH ENTROPY

INFORMATION CONTENT OF (X=x) = -LOG(P(X=x))

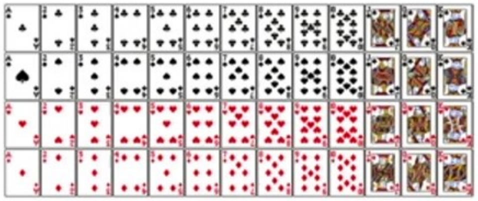
ENTROPY H(X) = AVERAGE VALUE OF THE INFORMATION CONTENT (ALSO CALLED THE EXPECTED VALUE)= $\sum p(X=x)\,(-\log(p(X=x))$

THE GAME IS TO ANSWER THE QUESTION

WHICH CARD DO YOU HOLD?

INITIALLY, THERE ARE 52 POSSIBLE OUTCOMES IN ALL

EACH HAS SAME PROBABILITY =1/52



ENTROPY=H(X)=

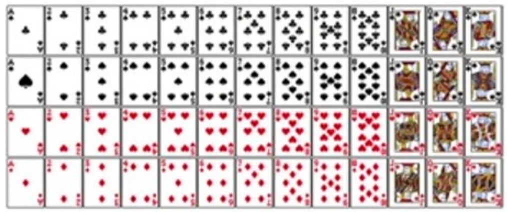
$\sum(1/52)(-\log(1/52)=$

LOG(52)

BEFORE WE HAVE ASKED ANY YES/NO QUESTIONS, THE UNCERTAINTY (ENTROPY) IN OUR GUESS IS VERY HIGH

THERE ARE 52 POSSIBLE OUTCOMES IN ALL

EACH HAS PROBABILITY =1/52



ENTROPY=H(X)= LOG(52)

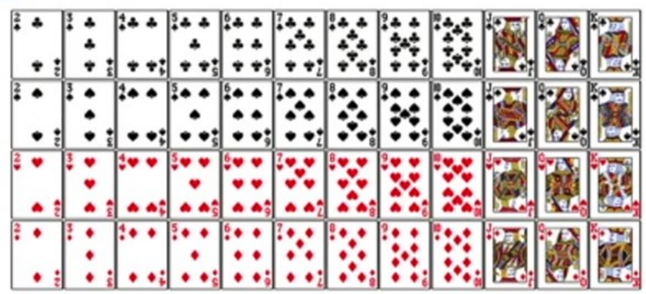
ONCE YOU ASK

IS THE CARD AN ACE?

YES

NO

ENTROPY = H(X/Q1=YES)= LOG(4)



BEFORE WE HAVE ASKED ANY YES/NO QUESTIONS, THE UNCERTAINTY (ENTROPY) IN OUR GUESS IS VERY HIGH

ENTROPY = $H(X)$ = $LOG(52)$

ONCE YOU ASK IS THE CARD AN ACE?

ENTROPY =	ENTROPY =
$H(X/Q1=YES)$=	$H(X/Q1=NO)$=
$LOG(4)$	$LOG(48)$
$P(YES)=4/52$	$P(NO)=48/52$

ENTROPY AFTER Q1 = $H(X/Q1)$ =

$P(YES)*H(X/Q1=YES)+P(NO)*H(X/Q1=NO)$ =

$4/52*\ LOG(4) + 48/52*LOG(48)$

BEFORE WE HAVE ASKED ANY YES/NO QUESTIONS, THE UNCERTAINTY (ENTROPY) IN OUR GUESS IS VERY HIGH

ENTROPY = H(X)

ONCE YOU ASK IS THE CARD AN ACE?

ENTROPY AFTER Q1 = H(X/Q1)

INFORMATION GAIN = REDUCTION IN ENTROPY OVERALL = H(X) - H(X/Q1)

AS YOU SAW, WHENEVER YOU ASK A QUESTION, SUBSETS ARE FORMED

WHEN EACH OF THOSE SUBSETS ARE HOMOGENOUS, THE INFORMATION GAIN IS MAXIMUM

-02:11

ENTROPY = H(X)

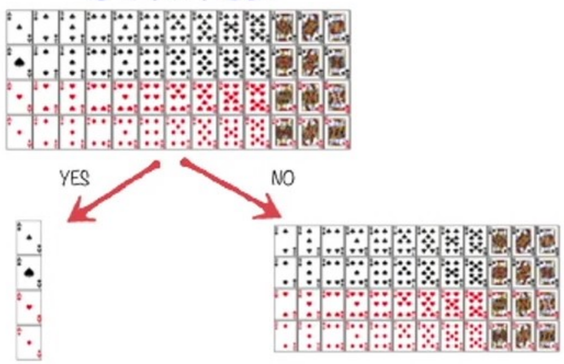
ENTROPY AFTER QI = H(X/QI)

INFORMATION GAIN = REDUCTION IN ENTROPY OVERALL = H(X) - H(X/QI)

AS YOU SAW, WHENEVER YOU ASK A QUESTION, SUBSETS ARE FORMED

WHEN EACH OF THOSE SUBSETS ARE HOMOGENOUS, THE INFORMATION GAIN IS MAXIMUM

IS IT AN ACE?



H(X/QI) = 4/52* LOG(4) + 48/52 * LOG(48)

IG= H(X)-H(X/QI) = 0.12

IS IT A BLACK?

H(X/QI) = LOG(26)

IG= H(X)-H(X/QI) = LOG(52)-LOG(26) =0.30

Decision Tree Learning

- Recursive partitioning is the most common strategy for decision tree learning
- Decision tree learning algorithms
  - CART
  - ID3
  - C4.5
  - CHAID

-01:14

Decision Tree Learning

- Recursive partitioning is the most common strategy for decision tree learning
- Decision tree learning algorithms
  - CART
  - ID3
  - C4.5
  - CHAID

EACH HAS A SLIGHTLY DIFFERENT WAY OF ARRIVING AT THE BEST ATTRIBUTE (OR) MEASURING THE HOMOGENEITY OF A SUBSET

Decision Tree Learning

- Recursive partitioning is the most common strategy for decision tree learning
- Decision tree learning algorithms
    - CART
    - ID3
    - C4.5
    - CHAID

INFORMATION GAIN

Decision Tree Learning

- Recursive partitioning is the most common strategy for decision tree learning
- Decision tree learning algorithms
  - CART
    GINI IMPURITY
  - ID3
  - C4.5
  - CHAID

CART IS ANOTHER DECISION TREE LEARNING METHOD

(CLASSIFICATION AND REGRESSION TREES)

IT USES A DIFFERENT WAY TO CHOOSE AN ATTRIBUTE

MINIMIZING GINI IMPURITY

THE IDEA BEHIND, GINI IMPURITY IS SIMPLE

CHOOSE THE ATTRIBUTE SUCH THAT - IF YOU STOP THE DECISION TREE WITH THAT ATTRIBUTE AND GO NO FURTHER

THE PROBABILITY OF A FALSE LABEL IS MINIMIZED

00:00

Key takeaways from this chapter

Building a Decision Tree - Information Gain a Gini Impurity

Entropy is the amount of Uncertainty/Unpredictability there is in the Answer.
The entropy typically changes when we use a node in a decision tree to partition the training instances into smaller subsets
Entropy increase with number of possible answers and the evenness of the probability distribution.
Entropy, $H(X) = \Sigma -P(E)Log2(P(E))$

Information gain is a measure of this change in entropy.
So in our case, it is the difference in entropy before the question is asked and after the question is asked.
Constructing a decision tree is all about finding an attribute that returns the highest information gain (i.e., the most homogeneous branches)

Key takeaways from this chapter

Decision Tree Lab:Building our First Decision Tree

1) import pandas and numpy libraries
2) Download Datset file:https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv
Even though the file comes with headers, we still specify them so that they are in our desired format
use read_csv and view the data
3)Since all the values are numeric we do not have to do encoding
4)import seaborn and matplotlib to plot he corelation heatmap
5) define x variable with all the available features except Quality and Y with just the labels
6)Use Split function to split using for training and testing.
7)Run the decision tree classifier in sklearn
8)print the accuracy of the trained model with respect to the test data

jupyter SVM-census Last Checkpoint: 14 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[43]:
```
import pandas as pd
import numpy as np
import math
```

[10]:
```
original_data=pd.read_csv("adult.csv",names =[ "Age", "Workclass", "fnlwgt", "Education", "Education-Num", "Marital Status", "Occupation", "Relationship",
                         sep=r'\s*,\s*',
                         engine='python',
                         na_values="?")
original_data.head()
#sep -- may or not have white space
```

[10]:

	Age	Workclass	fnlwgt	Education	Education-Num	Marital Status	Occupation	Relationship	Race	Gender	Capital Gain	Capital Loss	Hours per week	country	Target
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K

Key takeaways from this chapter

Decison Tree Lab: Viewing and Tweaking our Decision Tree

-You can use classifier.n_features_ attribute to view the number of features used in the decision tree.
-Tweak the max_depth attribute to observe the change in accuracy and minimise the size of the tree.

-You can use tree.export_graphviz() to output the decision tree as PNG file

-Tweak max_features to 4 has decreased the accuracy
-Specify Criterion As Entropy in order to use Information gain rather than Gini impurity to train the model and observe the acuracy against the test data.

Below are the 8 actual values of target variable in the train file.
[0,0,0,1,1,1,1,1]
What is the entropy of the target variable?

Options

- -(5/8 log(5/8) + 3/8 log(3/8))
- 5/8 log(5/8) + 3/8 log(3/8)
- 3/8 log(5/8) + 5/8 log(3/8)
- 5/8 log(3/8) 3/8 log(5/8)

For which of the following hyper parameters, higher value is better for decision tree algorithm?

1)Number of samples used for split

2)Depth of tree

3)Samples for leaf

Options

1 and 2

2 and 3

1 and 3

1, 2 and 3

Cant say

Following are the advantage/s of Decision Trees. Choose that apply.

Options

- Possible Scenarios can be added
- Use a white box model, If given result is provided by a model
- Worst, best and expected values can be determined for different scenarios
- All of the mentioned.
- can't say.

Decision Trees can be used for Classification Tasks

Options

- True
- False

Decision Tree is

Options

- Flow-Chart
- Structure in which internal node represents test on an attribute, each branch represents outcome of test and each leaf node represents class label
- Flow-Chart & Structure in which internal node represents test on an attribute, each branch represents outcome of test and each leaf node represents class label
- None of the mentioned

Movie Recommendation systems are not a case of:

Options

- Reinforcement learning
- Regression
- Clustering
- Classification

Sentiment analysis is a natural language processing (NLP) technique used to determine and analyze the sentiment expressed in a piece of text. It involves classifying the text into categories such as positive, negative, or neutral, and sometimes into more nuanced categories like joy, anger, or sadness.

Sentiment Analysis is not an example of:

Options

- Reinforcement learning
- Regression
- Clustering
- Classification