Chapter 4: Decision Trees Algorithms

Madhu Sanjeevi (Mady) · Follow
Published in Deep Math Machine learning.ai
6 min read · Oct 7, 2017

Listen Share

Decision tree is one of the most popular machine learning algorithms used all along, This story I wanna talk about it so let's get started!!!

Decision trees are used for both classification and regression problems, this story we talk about classification.

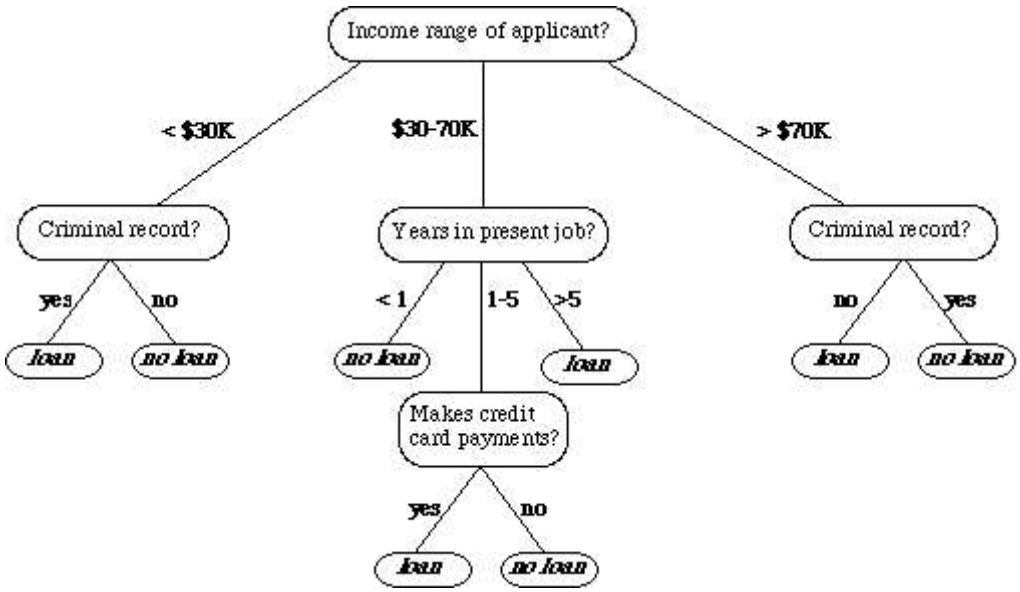
Before we dive into it , let me ask you this

> *Why Decision trees?*

We have couple of other algorithms there, so why do we have to choose Decision trees??

well, there might be many reasons but I believe a few which are

1. Decision tress often mimic the human level thinking so its so simple to understand the data and make some good interpretations.
2. Decision trees actually make you see the logic for the data to interpret(not like black box algorithms like SVM,NN,etc..)



For example : if we are classifying *bank loan* application for a customer, the decision tree may look like this

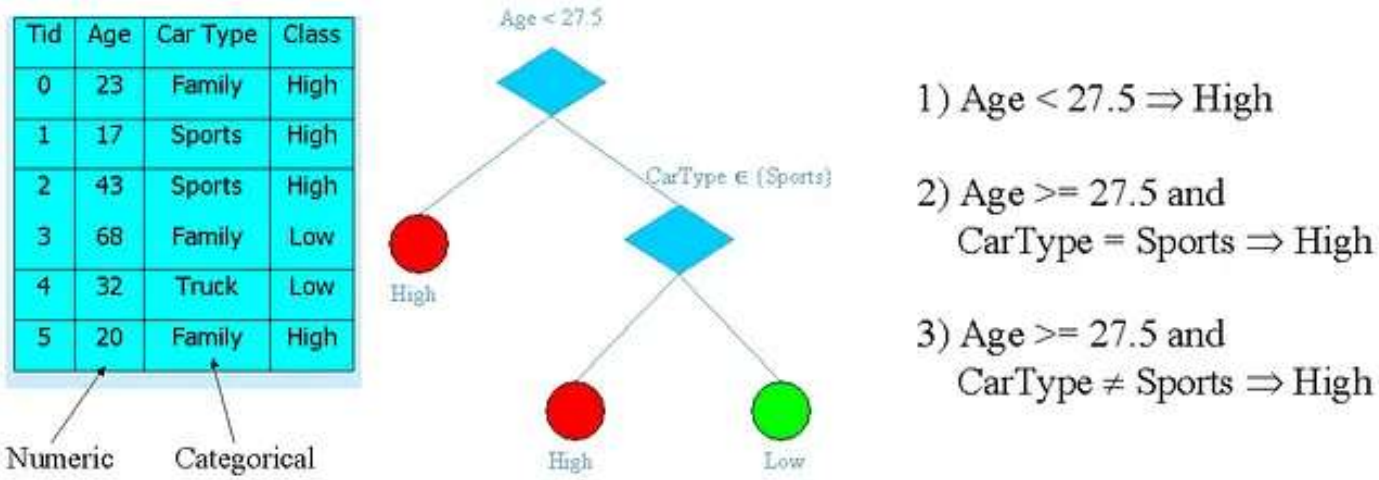
Here we can see the logic how it is making the decision.

It's simple and clear.

> *So what is the decision tree??*

A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).

The whole idea is to create a tree like this for the entire data and process a single outcome at every leaf(or minimize the error in every leaf).



Okay so how to build this??

There are couple of algorithms there to build a decision tree , we only talk about a few which are

1. CART (Classification and Regression Trees) → uses ***Gini Index(Classification)*** as metric.
2. ID3 (Iterative Dichotomiser 3) → uses ***Entropy function*** and ***Information gain*** as metrics.

Lets just first build decision tree for classification problem using above algorithms,

*Classification with using the **ID3** algorithm.*

Let's just take a famous dataset in the machine learning world which is weather dataset(playing game Y or N based on weather condition).

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

We have four X values (outlook,temp,humidity and windy) being categorical and one y value (play Y or N) also being categorical.

so we need to learn the mapping (what machine learning always does) between X and y.

This is a binary classification problem, lets build the tree using the ***ID3*** algorithm

To create a tree, we need to have a root node first and we know that nodes are features/attributes(outlook,temp,humidity and windy),

so which one do we need to pick first??

Answer: determine the attribute that best classifies the training data; use this attribute at the root of the tree. Repeat this process at for each branch.

This means we are performing top-down, greedy search through the space of possible decision trees.

okay so how do we choose the best attribute?

Answer: use the attribute with the highest ***information gain*** in ***ID3***

In order to define information gain precisely, we begin by defining a measure commonly used in information theory, called ***entropy*** *that characterizes the (im)purity of an arbitrary collection of examples."*

Entropy

Entropy $H(S)$ is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

Where,

- S – The current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
- C – Set of classes in S C={ yes, no }
- $p(c)$ – The proportion of the number of elements in class c to the number of elements in set S

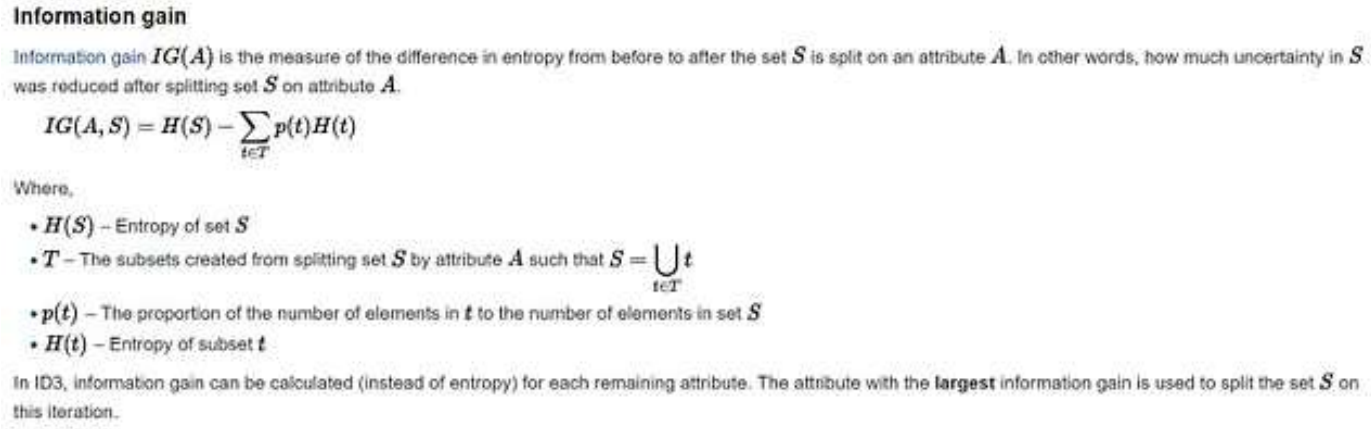
When $H(S) = 0$, the set S is perfectly classified (i.e. all elements in S are of the same class).

In ID3, entropy is calculated for each remaining attribute. The attribute with the **smallest** entropy is used to split the set S on this iteration. The higher the entropy, the higher the potential to improve the classification here.

wikipedia

For a binary classification problem

- If all examples are positive or all are negative then entropy will be *zero* i.e, low.
- If half of the examples are of positive class and half are of negative class then entropy is **one** i.e, high.



Information gain

Information gain $IG(A)$ is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in S was reduced after splitting set S on attribute A.

$$IG(A, S) = H(S) - \sum_{t \in T} p(t)H(t)$$

Where,

- $H(S)$ – Entropy of set S
- T – The subsets created from splitting set S by attribute A such that $S = \bigcup_{t \in T} t$
- $p(t)$ – The proportion of the number of elements in t to the number of elements in set S
- $H(t)$ – Entropy of subset t

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the **largest** information gain is used to split the set S on this iteration.

wikipedia

Okay lets apply these metrics to our dataset to split the data(getting the root node)

Steps:

```
1.compute the entropy for data-set

2.for every attribute/feature:
      1.calculate entropy for all categorical values
      2.take average information entropy for the current attribute
      3.calculate gain for the current attribute

3. pick the highest gain attribute.
4. Repeat until we get the tree we desired.
```

What the heck???

Okay I got it , if it does not make sense to you , let me make it sense to you.

Compute the entropy for the weather data set:

$$H(S) = \sum_{c \in C} -p(c)\log_2 p(c)$$

C ={yes,no}

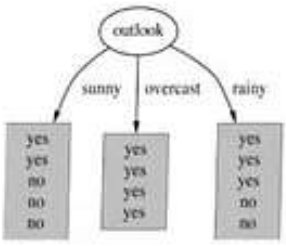
Out of 14 instances, 9 are classified as yes, and 5 as no

pyes = −(9/14)*log2(9/14) = 0.41

pno = −(5/14)*log2(5/14) = 0.53

H (S) = pyes + pno = 0.94

For every feature calculate the entropy and information gain



E (Outlook=sunny) = $-\frac{2}{5}\log\left(\frac{2}{5}\right)-\frac{3}{5}\log\left(\frac{3}{5}\right)=0.971$

E (Outlook=overcast) = $-1\log(1)-0\log(0)=0$

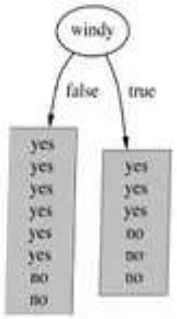
E (Outlook=rainy) = $-\frac{3}{5}\log\left(\frac{3}{5}\right)-\frac{2}{5}\log\left(\frac{2}{5}\right)=0.971$

} H(S,Outlook)

Average Entropy information for Outlook

I (Outlook) = $\frac{5}{14}*0.971+\frac{4}{14}*0+\frac{5}{14}*0.971=0.693$ } $\sum_{t\in T} p(t)H(t)$

Gain (Outlook) = E(S) − I (outlook) = 0.94-.693 = 0.247 ⇨ $IG(A,S)=H(S)-\sum_{t\in T}p(t)H(t)$



E (Windy=false) = $-\frac{6}{8}\log\left(\frac{6}{8}\right)-\frac{2}{8}\log\left(\frac{2}{8}\right)=0.811$

E (Windy=true) = $-\frac{3}{6}\log\left(\frac{3}{6}\right)-\frac{3}{6}\log\left(\frac{3}{6}\right)=1$

Average entropy information for Windy

I (Windy) = $\frac{8}{14}*0.811+\frac{6}{14}*1=0.892$

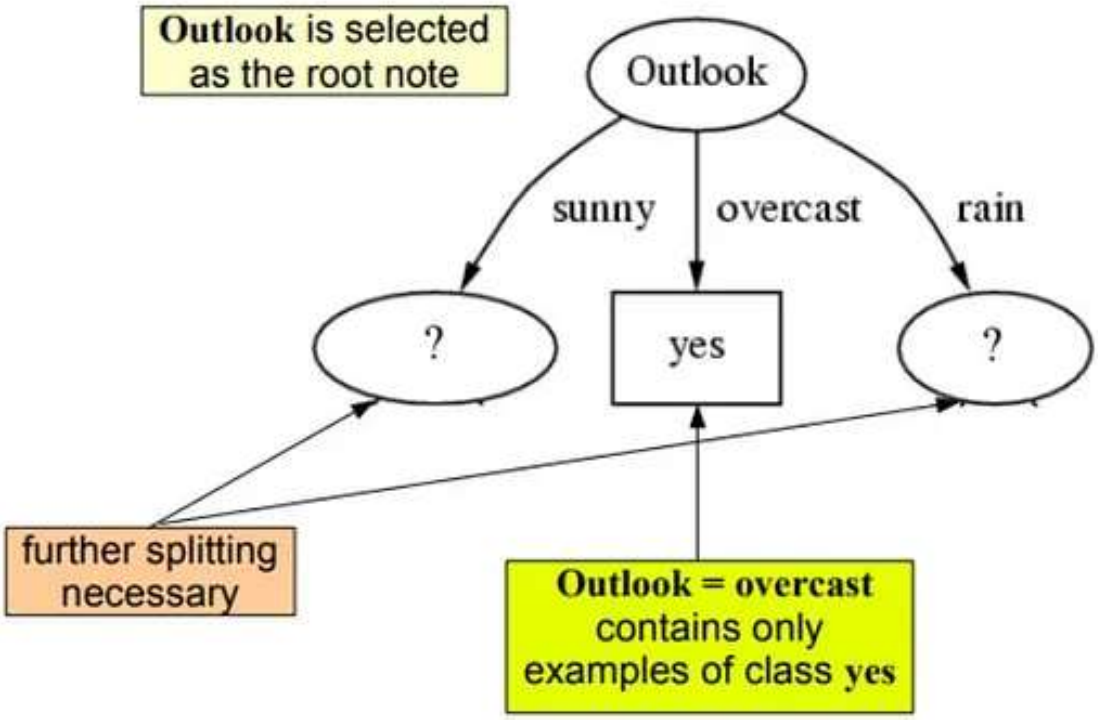
Gain (Windy) = E(S) − I (Windy) = 0.94-0.892=0.048

Similarity we can calculate for other two attributes(Humidity and Temp).

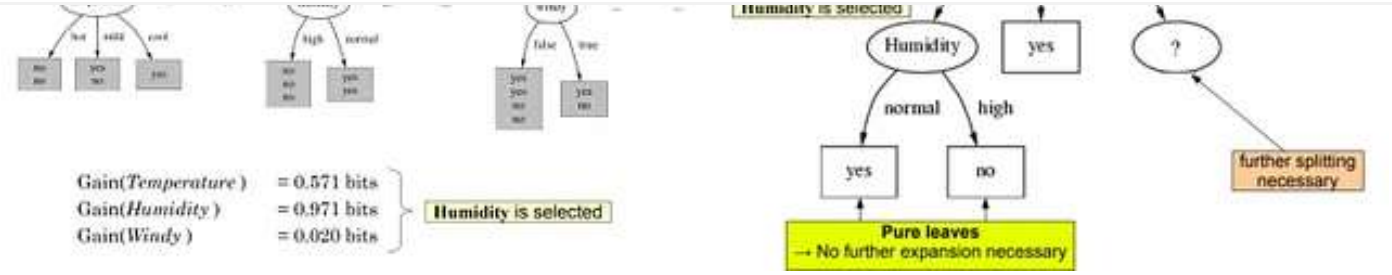
Pick the highest gain attribute.

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Humidity		**Windy**	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048

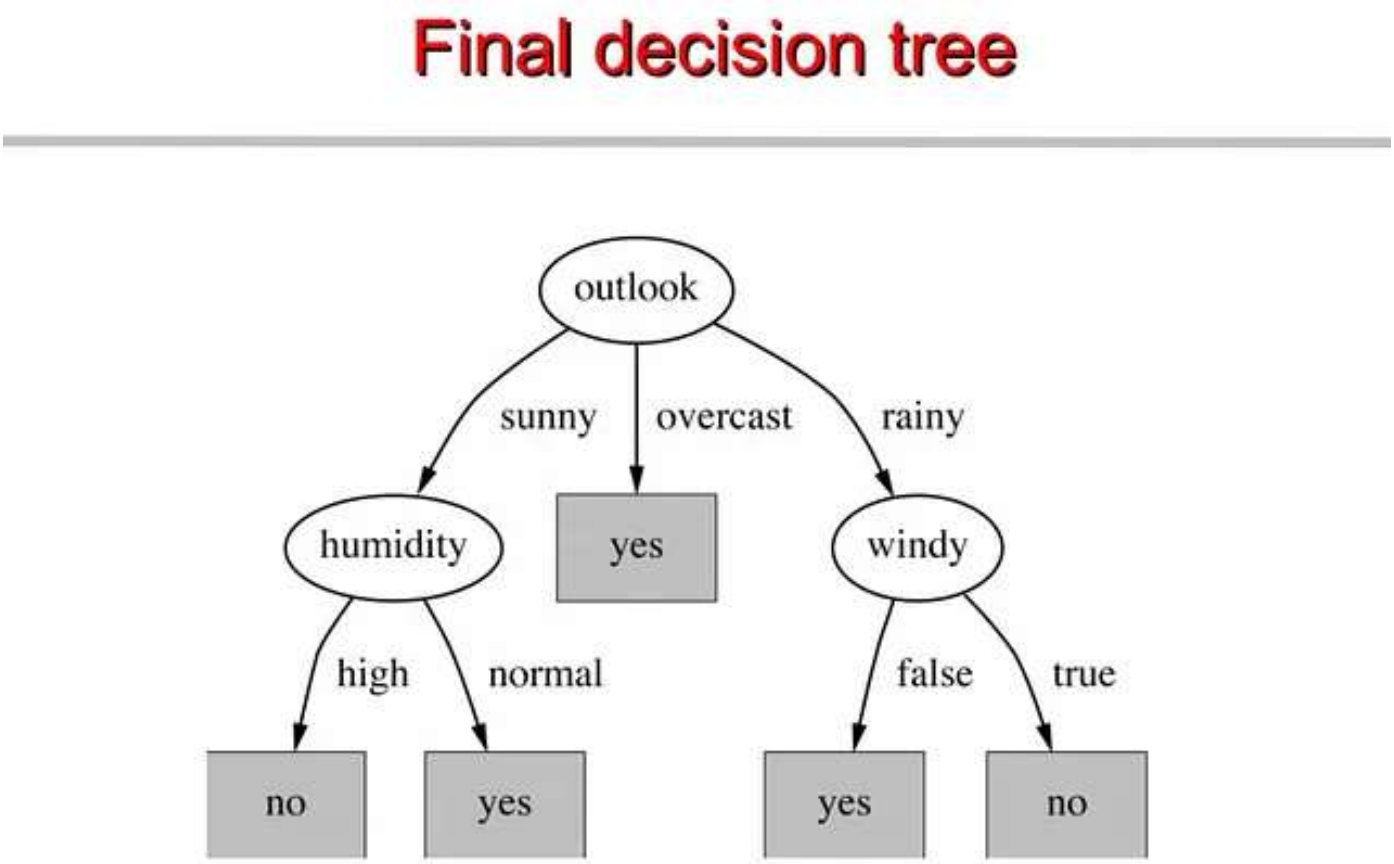
So our root node is **Outlook.**



Repeat the same thing for sub-trees till we get the tree.



Finally we get the tree something like his.



> *Classification with using the* ***CART*** *algorithm.*

In CART we use Gini index as a metric,

We use the Gini Index as our cost function used to evaluate splits in the dataset.

our target variable is Binary variable which means it take two values (Yes and No). There can be 4 combinations.

```
Actual=1 predicted 1
1 0 , 0,1, 0 0

P(Target=1).P(Target=1) + P(Target=1).P(Target=0) +
P(Target=0).P(Target=1) + P(Target=0).P(Target=0) = 1

P(Target=1).P(Target=0) + P(Target=0).P(Target=1) = 1 — P^2(Target=0)
— P^2(Target=1)
```

Gini Index for Binary Target variable is

= 1 — P^2(Target=0) — P^2(Target=1)

$$= 1 - \sum_{t=0}^{t=1} P_t^2$$

Gini index

A Gini score gives an idea of how good a split is by how mixed the classes are in the two groups created by the split. A perfect separation results in a Gini score of 0, whereas the worst case split that results in 50/50 classes.

We calculate it for every row and split the data accordingly in our binary tree. We repeat this process recursively.

For Binary Target variable, Max Gini Index value

= 1 — (1/2)^2 — (1/2)^2
= 1–2*(1/2)^2
= 1- 2*(1/4)
= 1–0.5
= 0.5

Similarly if Target Variable is categorical variable with multiple levels, the Gini Index will be still similar. If Target variable takes k different values, the Gini Index will be

$$1 - \sum_{t=0}^{t=k} P_t^2$$

Maximum value of Gini Index could be when all target values are equally distributed.

Similarly for Nominal variable with k level, the maximum value Gini Index is

= 1–1/k

Minimum value of Gini Index will be 0 when all observations belong to one label.

Steps:

```
1.compute the gini index for data-set

2.for every attribute/feature:
       1.calculate gini index for all categorical values
       2.take average information entropy for the current attribute
       3.calculate the gini gain

3. pick the best gini gain attribute.
4. Repeat until we get the tree we desired.
```

The calculations are similar to ID3 ,except the formula changes.

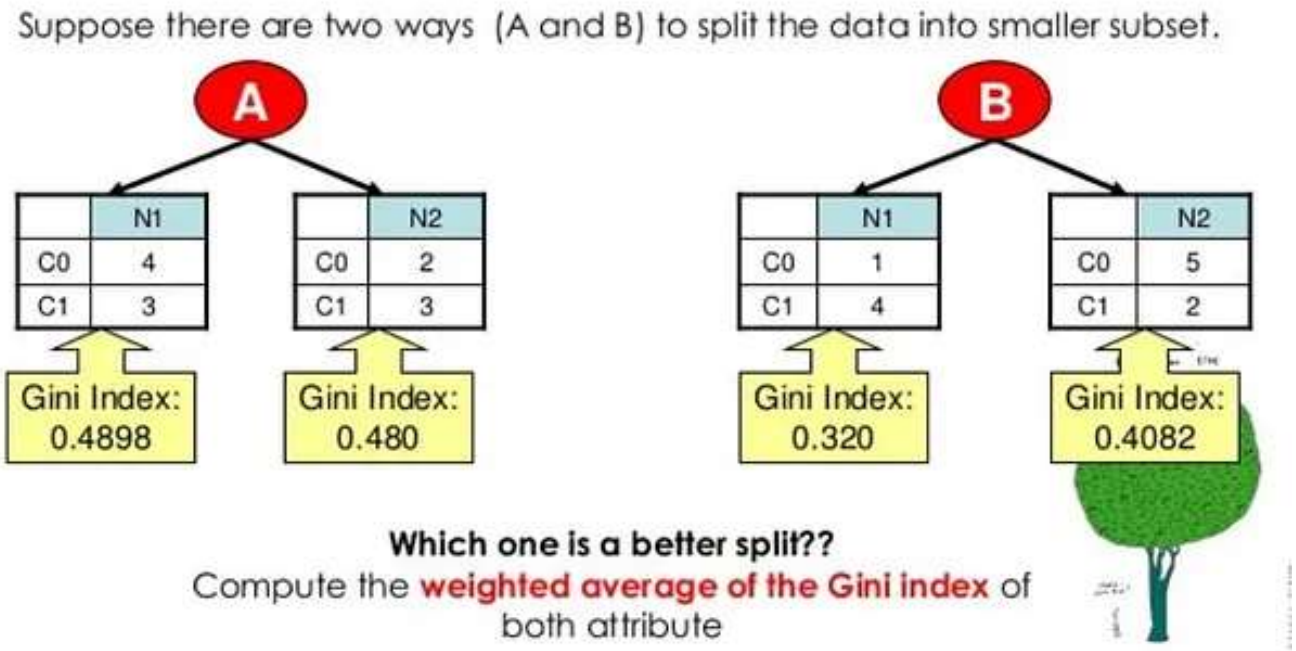
for example :compute gini index for dataset

$$= 1 - \sum_{t=0}^{t=1} P_t^2$$

Out of 14 instances ,
yes=9,no=5
1 - (9/14)^2- (5/14)^2

1-0.413-0.127= 0.46
Gini = 0.46

similarly we can follow other steps to build the tree



That's it for this story. hope you enjoyed and learned something.

Let me know your thoughts/suggestions/questions.

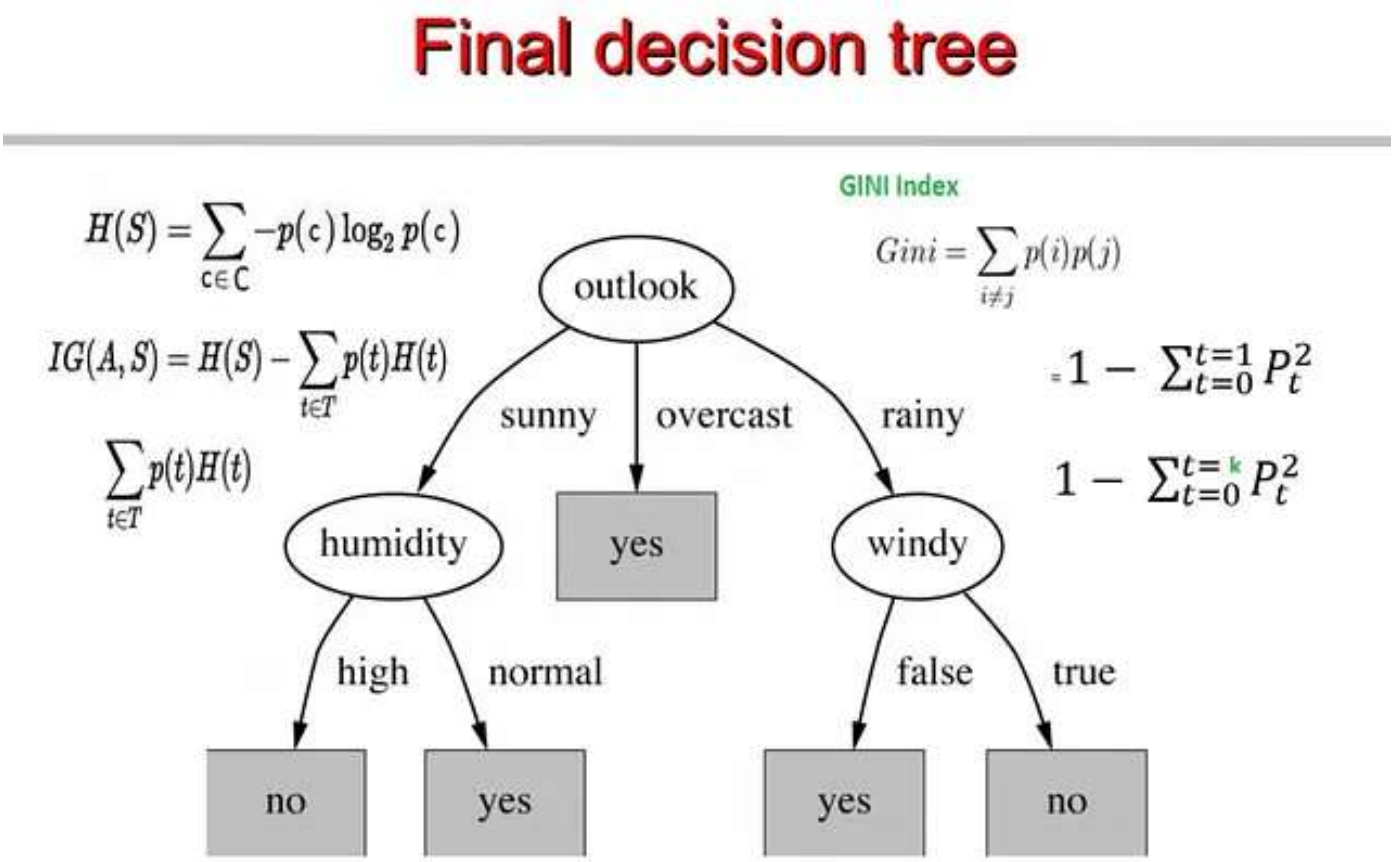
we just talked the first half of Decision trees , we can talk about the other half later (some statistical notations,theories and algorithms)

In the next story we will code this algorithm from scratch (without using any ML libraries).

Until then

See ya!

The images I borrowed from a pdf book which I am not sure and don't have link to add it. Let me know if anyone finds the abouve diagrams in a pdf book so I can link it.



Machine Learning Supervised Learning Decision Tree Entropy Classification

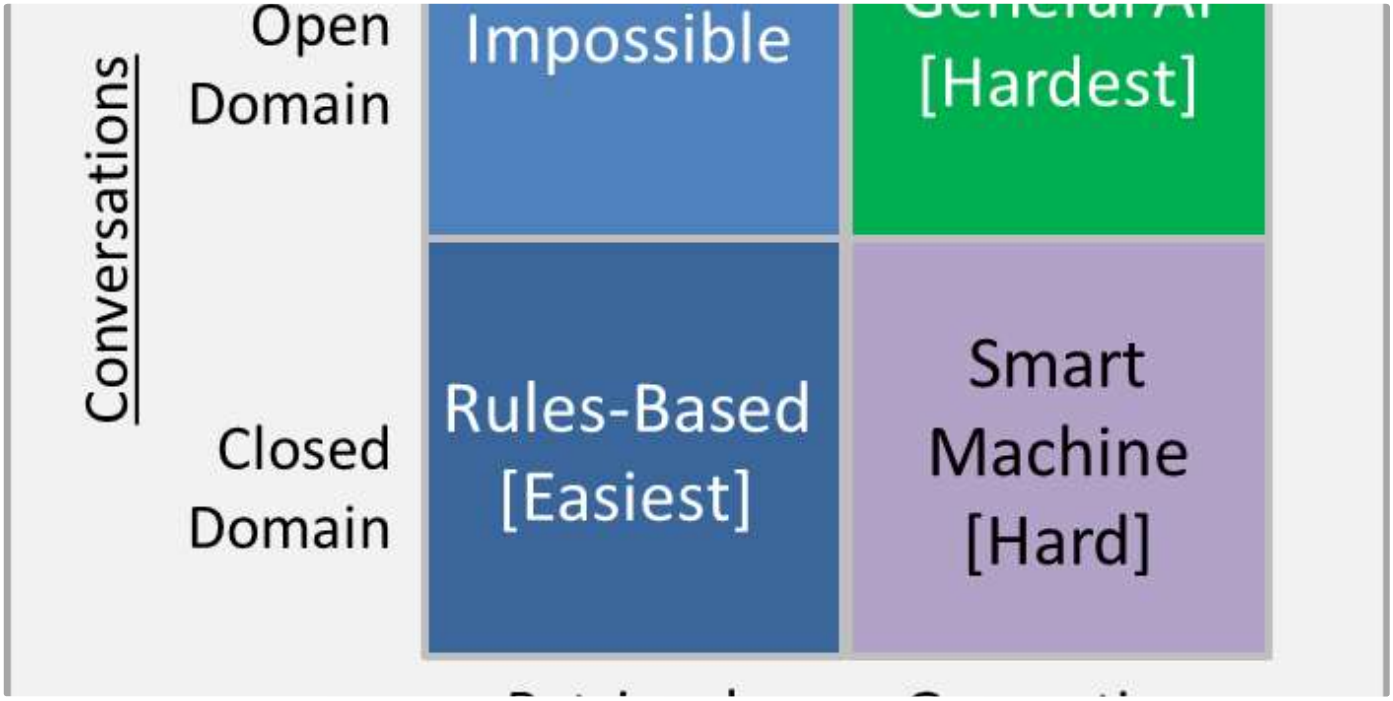
Follow

Written by Madhu Sanjeevi (Mady)

4.2K Followers · Editor for Deep Math Machine learning.ai

Writes about Technology (AI, Blockchain) | interested in Programming || Science || Math
https://www.linkedin.com/in/madhusanjeeviai

More from Madhu Sanjeevi (Mady) and Deep Math Machine learning.ai



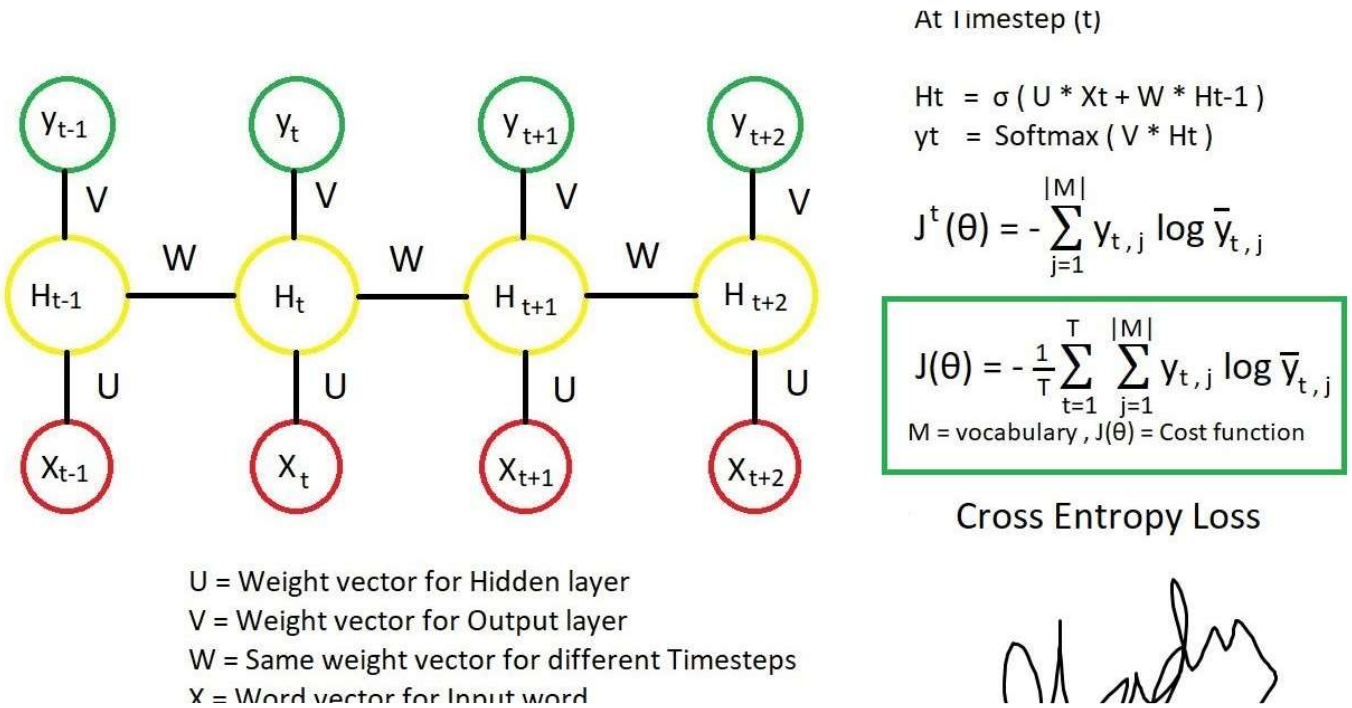
Madhu Sanjeevi (Mady) in Deep Math Machine learning.ai

Chapter 11: ChatBots to Question & Answer systems.

I am really excited to write this story , so far I have talked about Machine learning,deep learning,Math and programming and I am sick of...

13 min read · Apr 19, 2018

1.6K 18



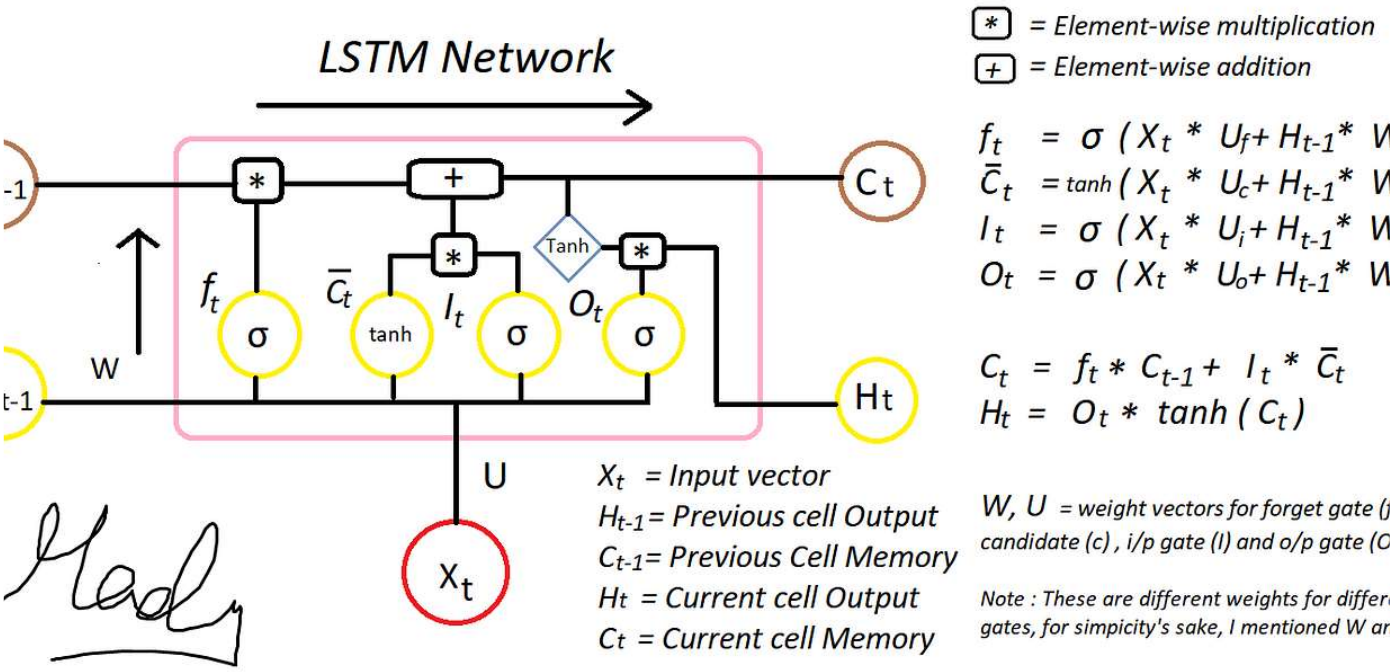
Madhu Sanjeevi (Mady) in Deep Math Machine learning.ai

Chapter 10: DeepNLP - Recurrent Neural Networks with Math.

we talked about normal neural networks quite a bit, Let's talk about fancy neural networks called recurrent neural networks.

6 min read · Jan 11, 2018

928 8



Madhu Sanjeevi (Mady) in Deep Math Machine learning.ai

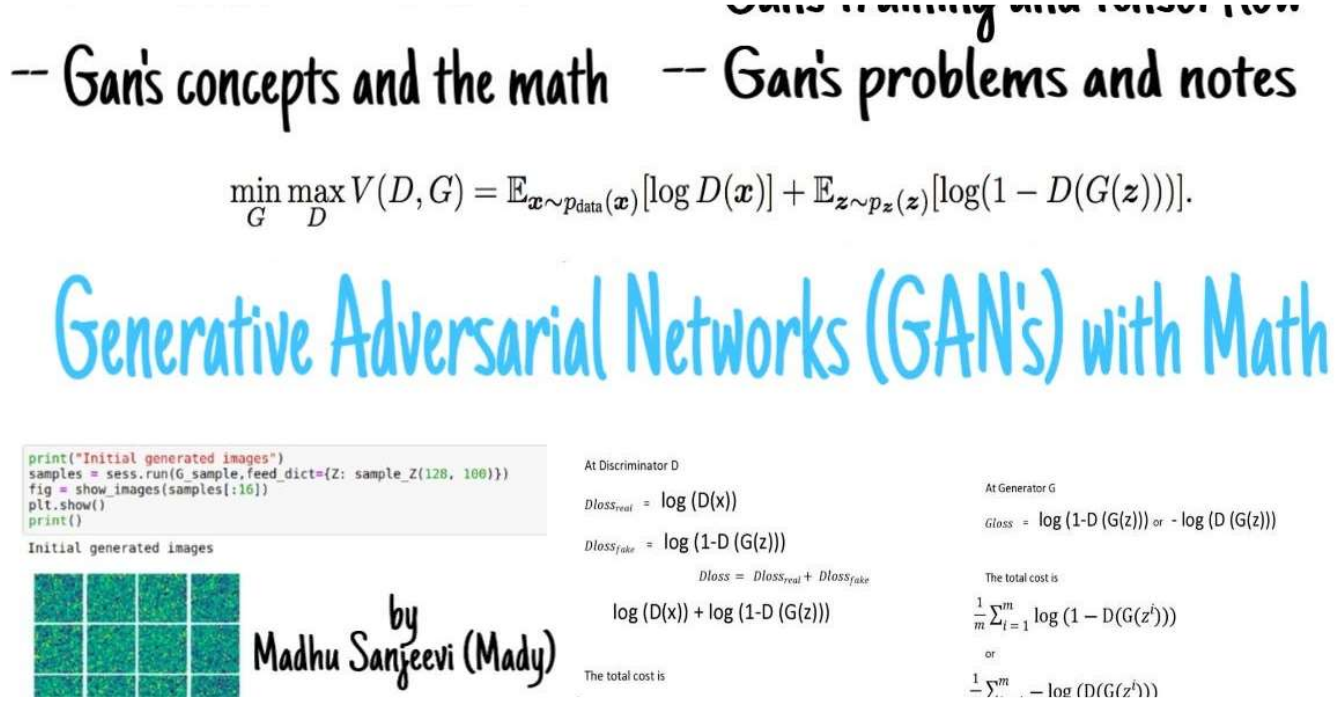
Chapter 10.1: DeepNLP — LSTM (Long Short Term Memory) Networks with Math.

Note: I am writing this article with the assumption that you know the deep learning a bit. In case if you don't know much, Please read my...

6 min read · Jan 21, 2018


961


6



Madhu Sanjeevi (Mady) in Deep Math Machine learning.ai

Ch:14 General Adversarial Networks (GAN's) with Math.

Discriminative vs generative , Gan's training and tensorflow, gan's concepts and the math and gans problems.

11 min read · Jan 14, 2019

Gencay I. in DataDrivenInvestor

Classification Task with 6 Different Algorithms using Python

Here are 6 classification algorithms to predict mortality with Heart Failure; Random Forest, Logistic Regression, KNN, Decision Tree, SVM...

· 9 min read · Nov 19, 2022

232 2

Md. Zubair in Towards Data Science

KNN Algorithm from Scratch

Implementation and Details Explanation of the KNN Algorithm

5 min read · Nov 16, 2022

370 1

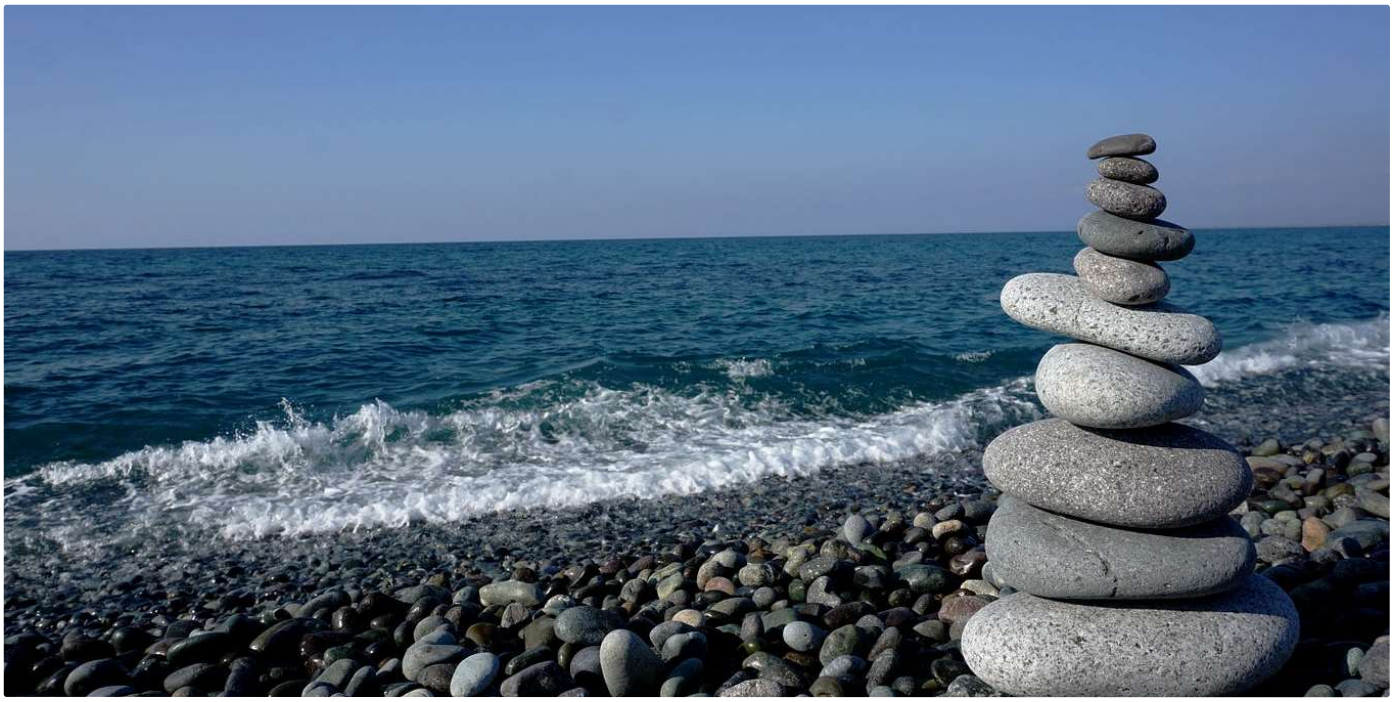
Lists

What is ChatGPT?

9 stories · 17 saves

Staff Picks

300 stories · 58 saves



Amy @GrabNGoInfo in GrabNGoInfo

Bagging vs Boosting vs Stacking in Machine Learning

Data Science Interview Questions and Answers

· 6 min read · Nov 24, 2022

Piero Paialunga in Towards Data Science

Naive Bayes Classifier from Scratch, with Python

From theory to practice with Bayes Theorem

✦ · 10 min read · Jan 4

75 3

Matt Chapman in Towards Data Science

The Portfolio that Got Me a Data Scientist Job

Spoiler alert: It was surprisingly easy (and free) to make

✦ · 10 min read · Mar 24

2.8K 43

Albers Uzila in Level Up Coding

Wanna Break into Data Science in 2023? Think Twice!

It won't be smooth sailing for you

✦ · 11 min read · Dec 23, 2022

828 13