[EN] Visualising Tree Data with ETE Toolkit

Leaf it to me to help you branch out

Andrew Hetherington · Follow
Published in Towards Data Science · 6 min read · May 24, 2020

The world is very complex. Things are related in all kinds of different ways — some relations are intuitive, some aren't. As a result, the data that we collect and use to describe the world comes in all kinds of shapes and sizes.

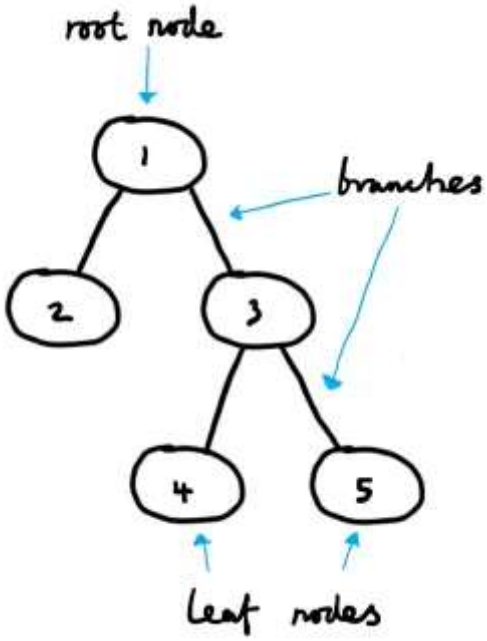
In addition, Data Visualisation is one of the most important aspects of the data analysis process. *Good visualisation is critical in helping clients and stakeholders understand data.*

In combination, these two points mean that people who work with data need to be prepared to represent it in ways that *best illustrate the relationships* between data points — and *what it might mean for action*. Today, we're going to look at one such representation and a tool you can use to create it.

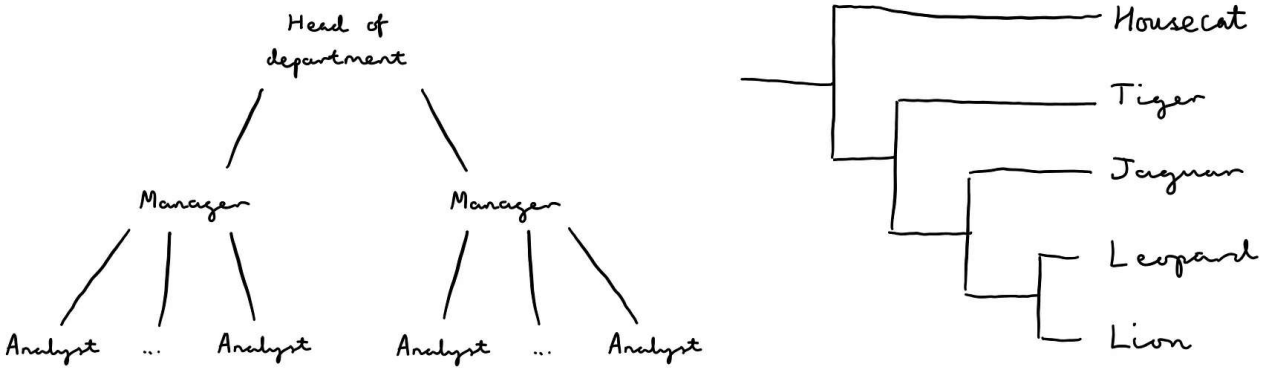
Photo by Todd Quackenbush on Unsplash

Trees as a data structure

At work, I recently came into contact with *trees*. A *tree* is a way of representing a set of hierarchical data. The relations between instances in a tree are fairly intuitive to understand, thanks to the ready analogy with the botanical trees that we are all familiar with. The ideas of roots, leaves and branches transfer over quite well when we are talking about the data structure rather than the perennial plant.



Trees and hierarchies are found all over the place — common real-world examples are company structures and evolutionary trees.



As you can see, it's relatively easy to sketch out a tree for a given set of hierarchically linked data. But what about when you have large amounts of data, or your data is constantly evolving? Redrawing the whole thing by hand is painstaking and time-consuming...

ETE Toolkit

…And that's where ETE Toolkit comes in. This is a Python library for the analysis and visualisation of trees. It was originally developed by the bioinformatics department at the Principe Felipe Research Center in Valencia, Spain, so it's capable of all kinds of scientific analysis. It's also a handy way to automate the visualisation of any tree data structure, the basics of which I'm going to describe here. Check out ETE Toolkit's Gallery to see some of the package's more advanced features.

Visualising a customer referral system

Let's use a customer referral system as an example. You're probably familiar with how this works — a customer who purchases a product is given a unique code that they can share with their friends. If one of their friends decides to also purchase the product and quotes this unique code when doing so, both earn a reward. It's a win-win scenario — the company selling the product is able to align the customer's interests with its own by incentivising them to spread the word and encourage more people to become customers — and of course, the customers themselves enjoy a discount.

Say each customer is allowed to refer up to 5 people. Say also that we currently have a tabular representation of this data as follows:

	Date	Name	Referred by	Amount spent (£)
0	18/05/2020	Alice	NaN	4.35
1	18/05/2020	Bob	Alice	28.00
2	18/05/2020	Carol	Bob	3.70
3	19/05/2020	Niaj	Alice	4.50
4	19/05/2020	Mallory	Bob	3.30
5	19/05/2020	Dan	Carol	5.30
6	19/05/2020	Erin	Carol	9.00
7	19/05/2020	Frank	Dan	5.00
8	19/05/2020	Grace	Erin	5.00
9	20/05/2020	Heidi	Frank	3.00

We have data on a number of purchases, including:

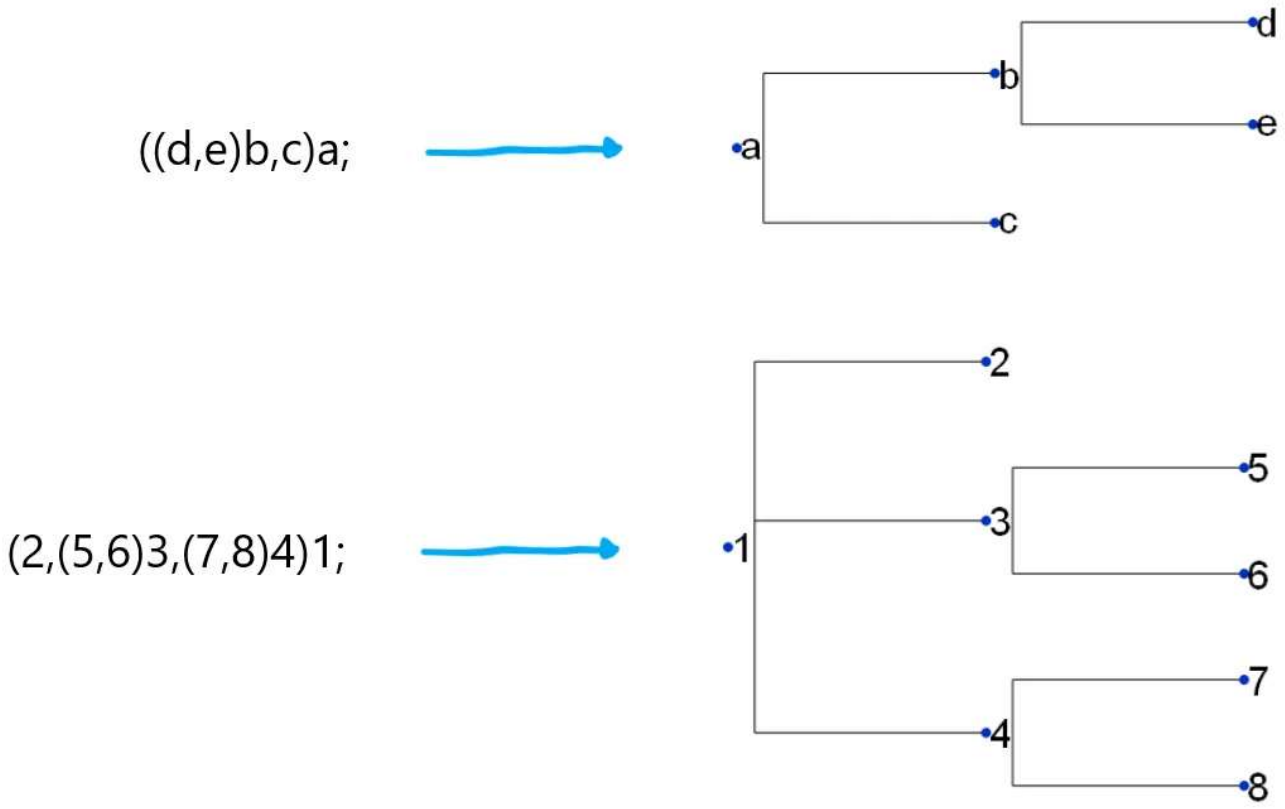
- date of transaction;
- name of customer;
- who referred the customer; and
- how much they spent.

This data forms a hierarchy — each customer can be represented as a *node* in a tree diagram, with the person who referred them placed above them in the tree, and the people they went on to refer branching off below them. We can also see here that Alice is the "original" customer that has started this particular chain of referrals — that is, Alice is the only customer who was not referred by someone else. This makes her the *root* customer.

Given this, we could draw a tree diagram by hand — going through each customer in turn, working out who should be linked to them in the tree, and

adding the appropriate as *children,* branching off underneath them. But why do that when ETE will gladly to do the hard work for us?

ETE takes input in *Newick format* — a relatively standard way of compactly describing the layout of tree structures in mathematics and science. Here's a couple of examples:



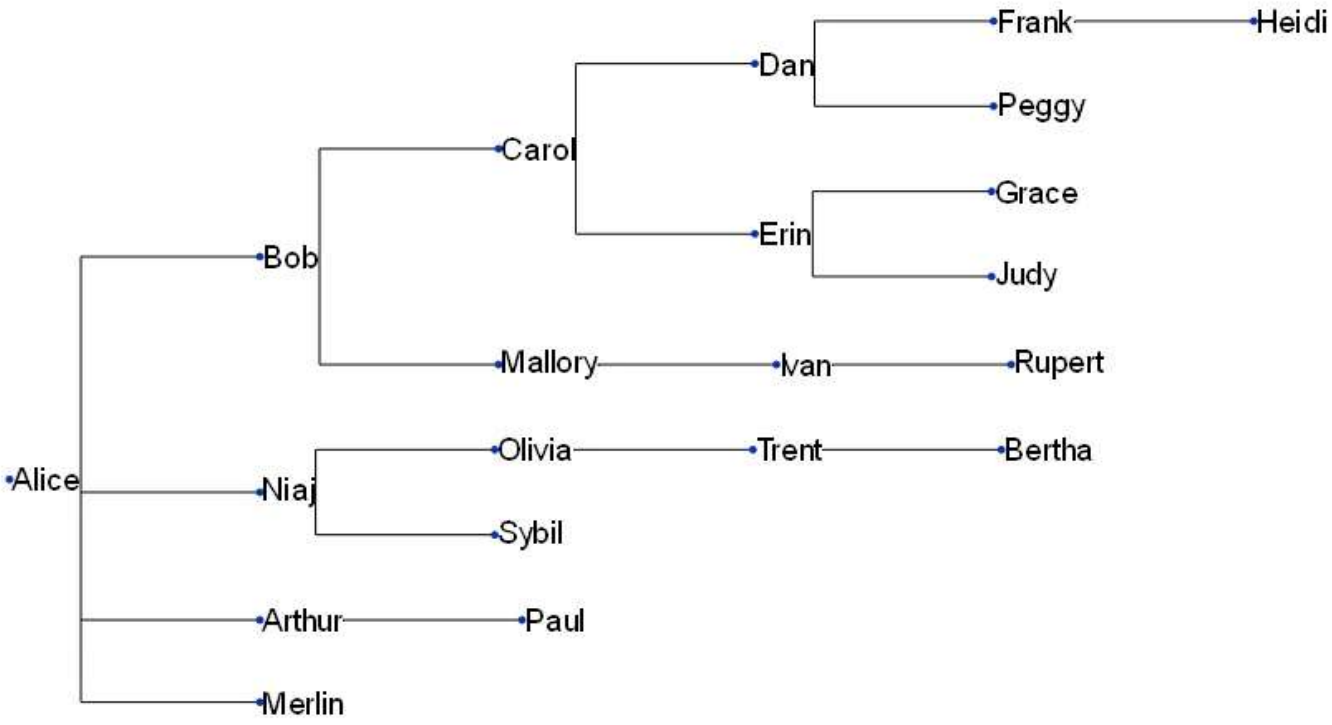
You may be able to notice that a set of branches is represented by a pair of parentheses. The number of branches coming off of a given node is determined by the presence of commas within the parentheses. And of course, pairs of parentheses can be nested to produce more and more complex structures.

I'm not going to spend too long on the details of Newick format — it's sufficient to know that we can tell ETE what we want to visualise by converting our data into this format. I've written some code to do this for us, which I won't describe here — if you're interested, take a look at the Jupyter Notebook on GitHub that goes with this post for more information.

Our customer data can thus be written in Newick format as follows:

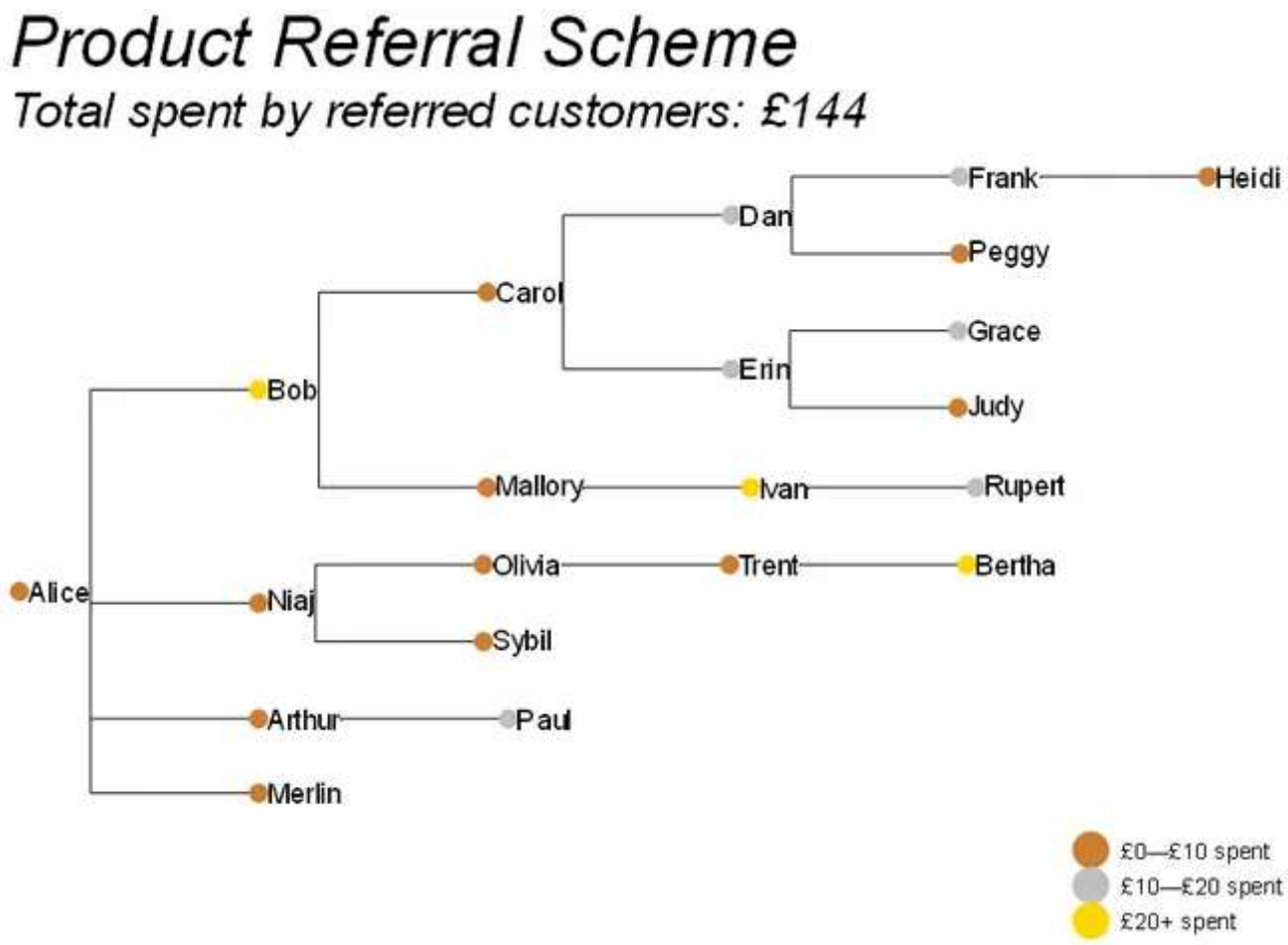
```
(((((Heidi)Frank,Peggy)Dan,(Grace,Judy)Erin)
Carol,((Rupert)Ivan)Mallory)Bob,(((Bertha)Trent)
Olivia,Sybil)Niaj,(Paul)Arthur,Merlin)Alice;
```

Let's see what we get when we feed it into ETE:



Not too bad! Looking quite basic at the moment — but ETE has taken a lot of the complexity out of the visualisation process. What's more, we can simply re-run our code whenever we get updated data!

Another advantage of ETE is that we can quite easily make adjustments to presentational aspects of the tree. Let's see if we can make it look a bit snazzier...



That's better. We're still just scratching the surface of what ETE Toolkit is capable of, but as a quick illustration, this is enough to get some of the key ideas across.

Conclusion

The human brain loves to hoover up visual information. Trees have a ready real-world analogy that makes them intuitive and easy to understand. This

data when it is presented to us in a way that works *with* these peculiarities, rather than *against* them.

More info and credits

Andrew Hetherington is an actuary-in-training and data enthusiast based in London, UK.

- Connect with me on LinkedIn.
- See what I'm tinkering with on GitHub.
- The notebook used to produce the plots in this article can be found here.

Tree photo by Todd Quackenbush on Unsplash.

ETE 3: Reconstruction, analysis and visualization of phylogenomic data.

Jaime Huerta-Cepas, Francois Serra and Peer Bork.

Mol Biol Evol 2016; doi: 10.1093/molbev/msw046

Data Science | Data | Data Visualization | Data Analysis | Data Structures

Written by Andrew Hetherington

50 Followers · Writer for Towards Data Science

Follow

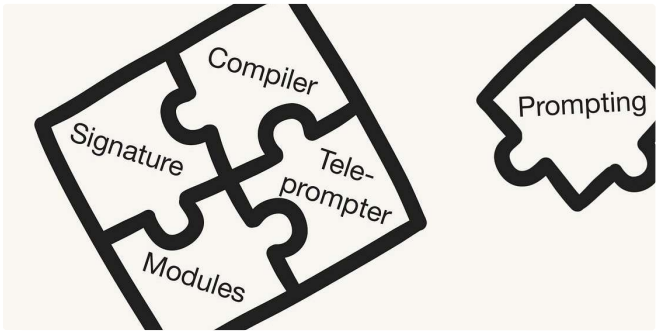
More from Andrew Hetherington and Towards Data Science

Andrew Hetherington in Towards Data Science

Maximum Likelihood Estimation in R

Maximise your likelihood of statistical success with this quick and easy guide

6 min read · Jul 20, 2020



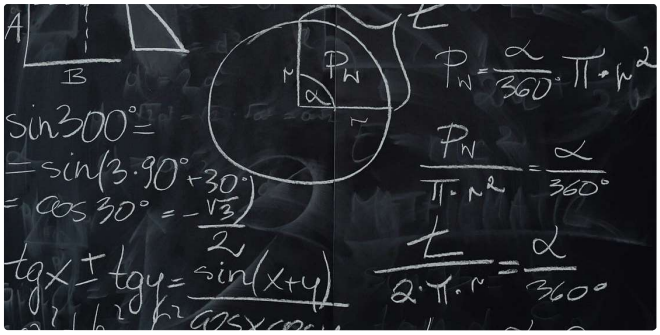
Leonie Monigatti in Towards Data Science

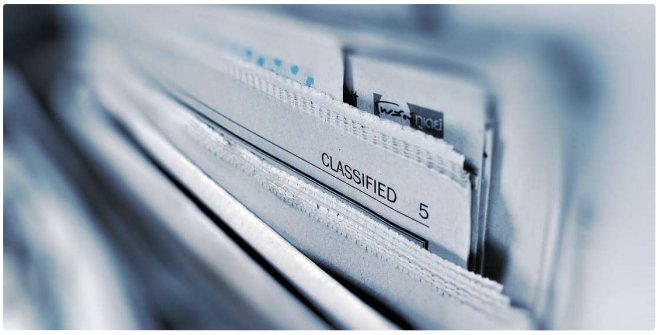
Intro to DSPy: Goodbye Prompting, Hello Programming!

How the DSPy framework solves the fragility problem in LLM-based applications by...

13 min read · Feb 27, 2024

110 3.5K 10





Egor Howell in Towards Data Science

How to Learn the Math Needed for Data Science

A breakdown of the three fundamental math fields required for data science: statistics,...

✨ · 8 min read · Mar 5, 2024

1.6K 9

Andrew Hetherington in Towards Data Science

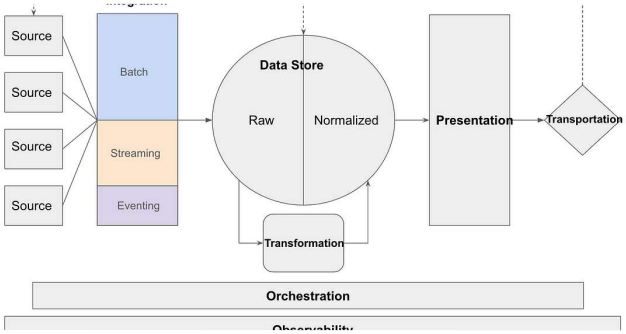
Evaluating Classifier Model Performance

Precision, Recall, AUC and more — demystified

17 min read · Jul 5, 2020

17

Recommended from Medium



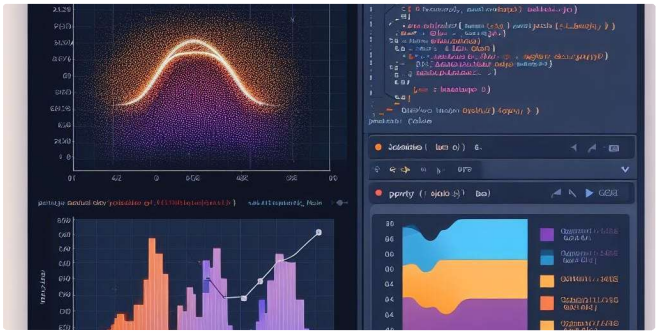
Dave Melillo in Towards Data Science

Building a Data Platform in 2024

How to build a modern, scalable data platform to power your analytics and data science...

9 min read · Feb 6, 2024

2.5K 35



Daniel Wu

Elevate Your Python Data Visualization Skills: A Deep Dive...

Data visualization is a crucial aspect of data analysis and exploration. It helps in gaining...

8 min read · Nov 22, 2023

343 1

Lists

Predictive Modeling w/ Python

20 stories · 1034 saves

Practical Guides to Machine Learning

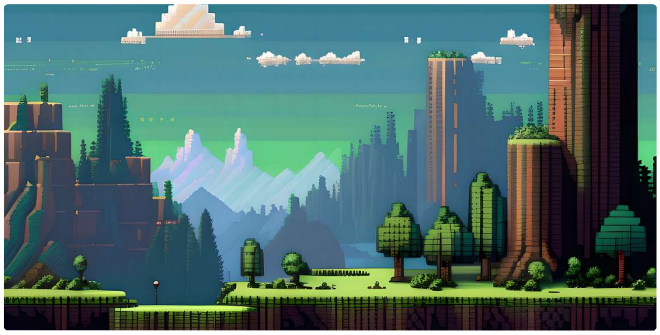
10 stories · 1236 saves

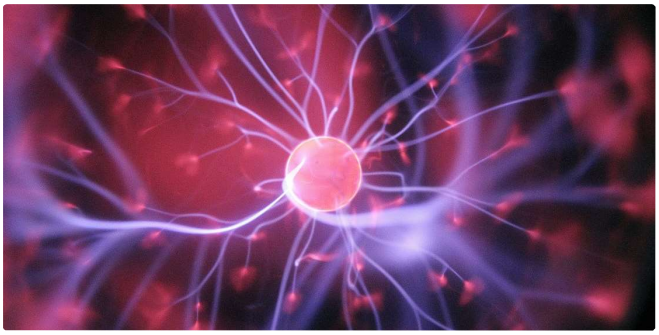
ChatGPT prompts

47 stories · 1329 saves

Coding & Development

11 stories · 522 saves





Furcy Pin in Learning SQL

Writing SQL Like A Pro: Advanced Techniques Showcased in a Real-...

Merging two history tables using advanced SQL techniques with Google's BigQuery.

9 min read · Mar 6, 2024

634 8

Rosaria Silipo in Low Code for Data Science

Is Data Science dead?

In the last six months I have heard this question thousands of time: "Is data science...

6 min read · Mar 11, 2024

1K 23

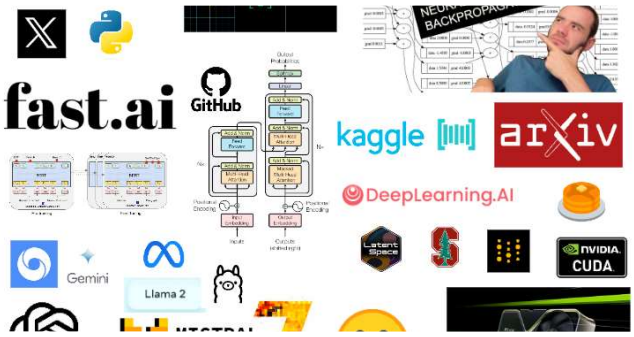
Anmol Tomar in CodeX

Say Goodbye to Loops in Python, and Welcome Vectorization!

Use Vectorization—a super-fast alternative to loops in Python

· 5 min read · Dec 28, 2023

4.96K 60



Benedict Neo in bitgrit Data Science Publication

Roadmap to Learn AI in 2024

A free curriculum for hackers and programmers to learn AI

11 min read · Mar 11, 2024

7.9K 84