Writing and Scaling Collaborative Data Pipelines with Kedro

2020

QUANTUMBLACK
A MCKINSEY COMPANY

Tam Nguyen

ARCHITECT AND DATA ENGINEER

/ BACKGROUND

Data Engineering and Architecture at various startups in many industries and verticals around the world over the past decade.

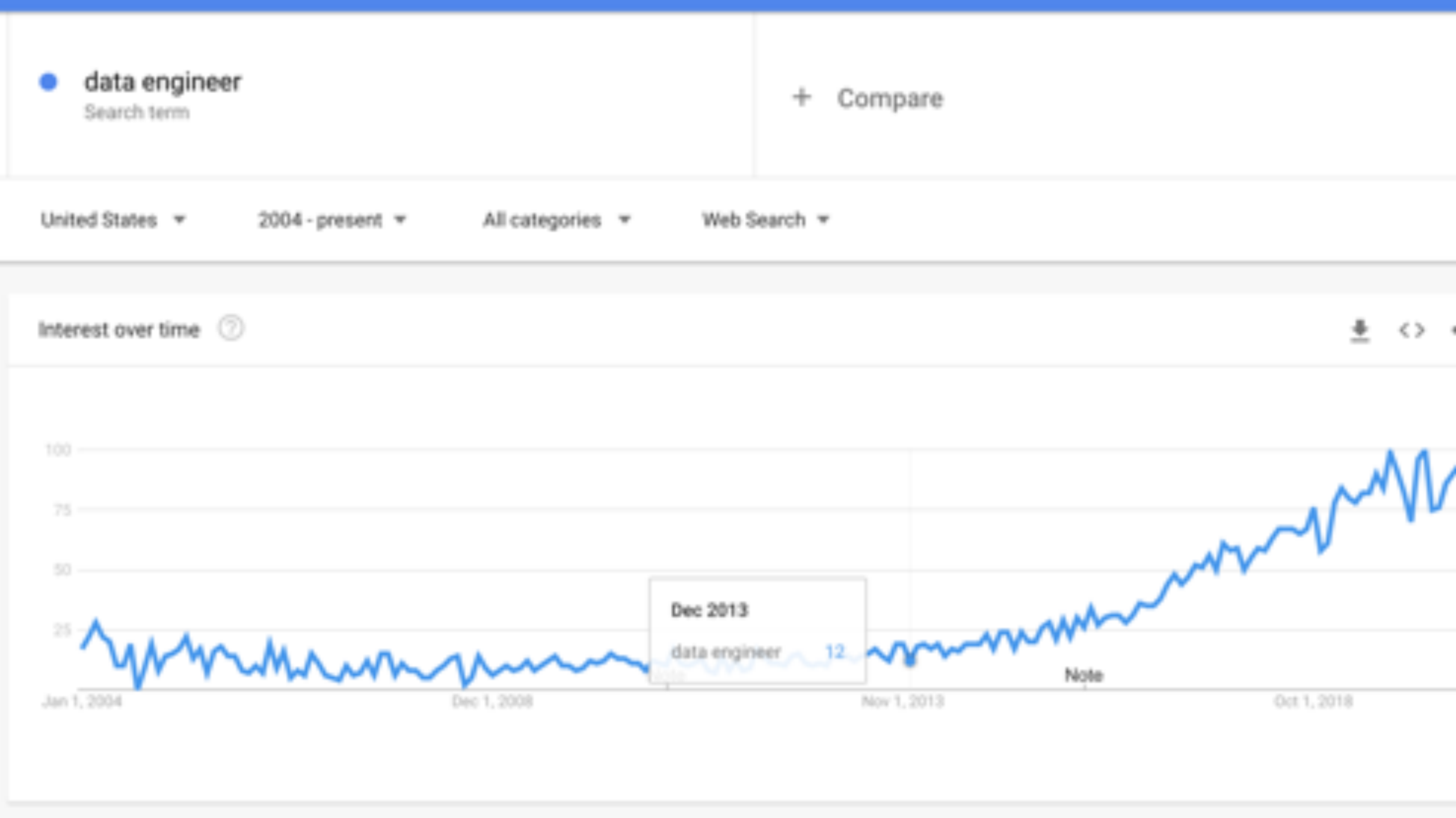
/ HOBBIES

Running a YouTube channel dedicated to Data Engineering. Teaching meditation and yoga to my friends and colleagues.

/ EDUCATION

University of Maryland, College Park
Computer Science

Let’s Talk about Pipelines



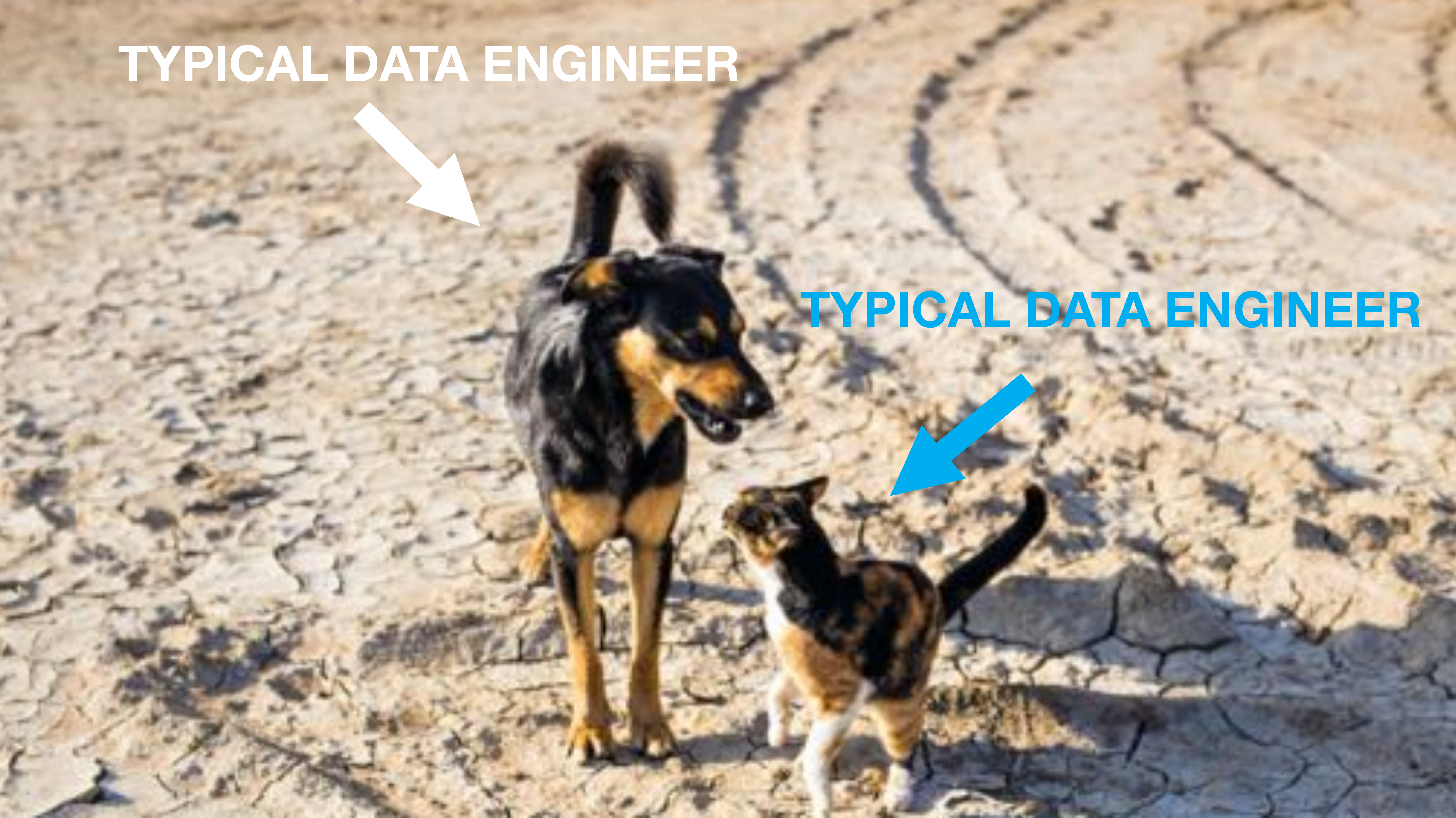
● data engineer

Search term

+ Compare

United States ▾ 2004 - present ▾ All categories ▾ Web Search ▾

Interest over time



TYPICAL DATA ENGINEER

TYPICAL DATA ENGINEER

DATA SCIENCISTS MAY NOT BE ENGINEERS

DATA SCIENCE TENDS TOWARDS EXPERIMENTATION

DATA ENGINEERS MAY NOT BE SCIENTISTS

DATA ENGINEERING TENDS TOWARD ORDER

BOTH MUST CLEAN

IS IT READY FOR THE HAND OFF?

IS IT PRODUCTION READY?

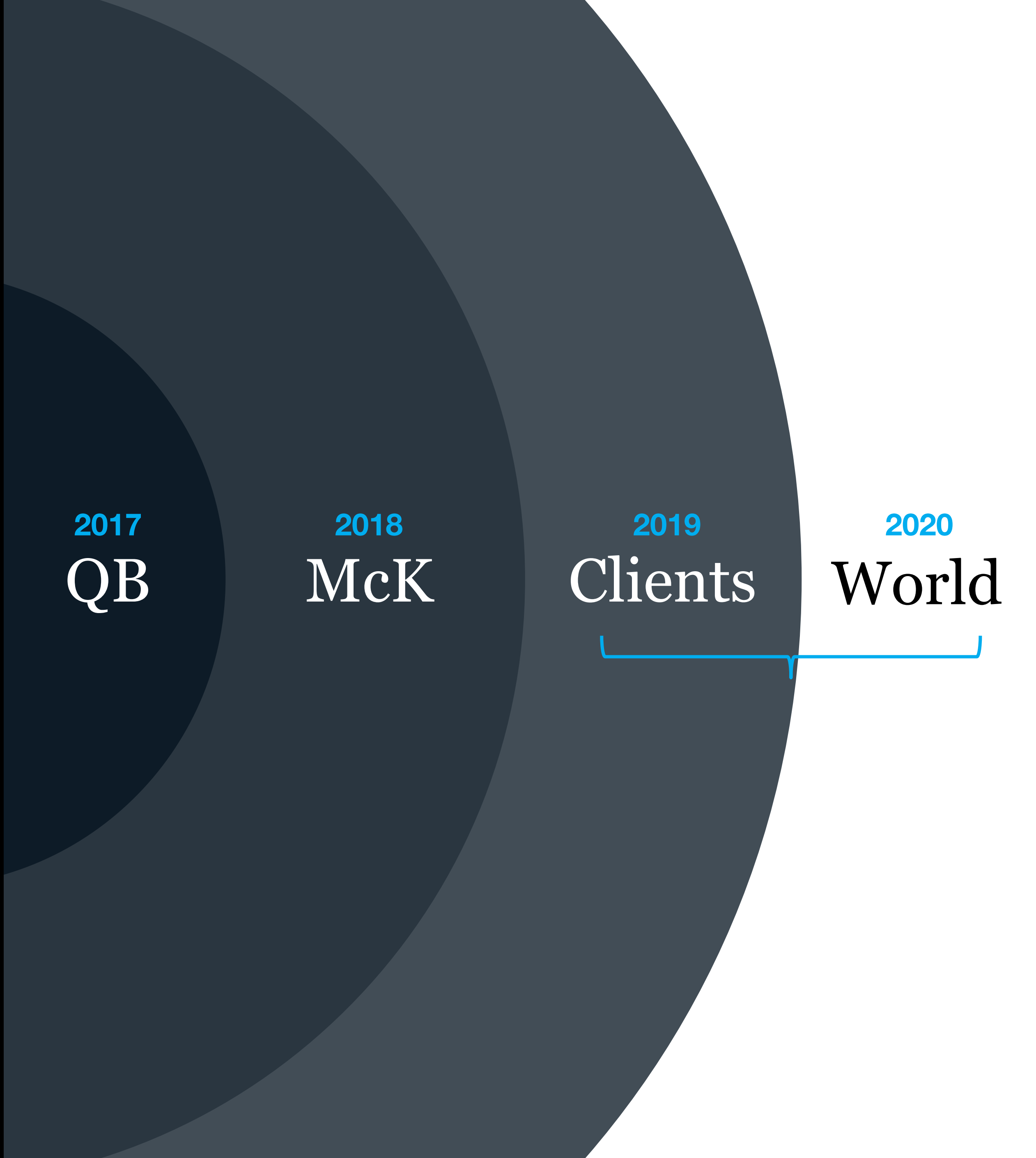
CAN WE FIND BALANCE?

How did QuantumBlack address these problems?

CREATION AND USE OF KEDRO

Kedro was the brain child of our projects, all over the world. We found that there were always these similar patterns we could constantly reuse.

> *"We're really glad that you open sourced Kedro because it's made it possible for Kedro to be used in all of our internal data science projects regardless of size."*
>
> - Client, Senior Data Scientist



Why does Kedro exist?

Kedro is built upon our collective best-practice (and mistakes) trying to deliver real-world ML applications that have vast amounts of dirty data.

Product Mission

Our teams come from many different backgrounds with varying experience with software engineering principles. It's with empathy that we say, *"How can we tweak your workflow so that our coding standards are the same?"*

Objectives

PRIMARY

A successful project does not only entail having a model run in production; our success is a client that can maintain their own data pipeline when we leave.

SECONDARY

We have time to do code and model optimization but we do not have time to refactor code. This means that we needed a seamless way to quickly move from the experimentation phase into production-ready code.

How does Kedro solve these problems?

IMAGINE AUDIO AS DATA

STANDARDIZED INPUTS/OUTPUTS

FUNCTIONAL TRANFORMERS

REDIRECTING COMPONENTS

CONVENTION FOR ORGANIZATION

The Catalog (Standardized Inputs and Outputs)

INTEGRATIONS IN THE CATALOG

pandas	Pandas
APACHE Spark	Spark
DASK	Dask
SQLAlchemy	SQLAlchemy
NetworkX	NetworkX
matplotlib	MatplotLib
Google BigQuery	Google BigQuery
Google Cloud Storage	Google Cloud Storage
amazon REDSHIFT	AWS Redshift
amazon S3	AWS S3
Microsoft Azure Blob Storage	Azure Blob Storage
hadoop	Hadoop File System

What is the catalog?

- Manages the loading and saving of your data
- Available as a code or YAML API
- Versioning is available for file-based systems every time the pipeline runs
- It’s extensible, and we accept new data connectors

What does configuration help you do?

- ✓ Never write a single line of code that would read or write to a file, database or storage system
- ✓ Makes it possible to write generalizable and reusable analytics code that does not require significant modification to be used
- ✓ Access data without leaking credentials

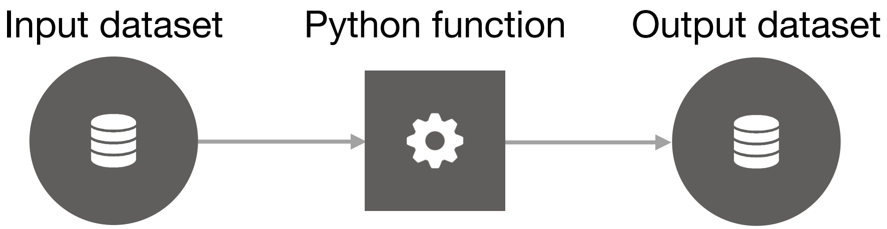
Nodes & Pipelines (Functional Transformers and Redirecting Components)

What are nodes?

- Usually a pure Python function that has an input and an output.
- Node definition supports multiple inputs for things like table joins and multiple outputs for things like producing a train/test split.

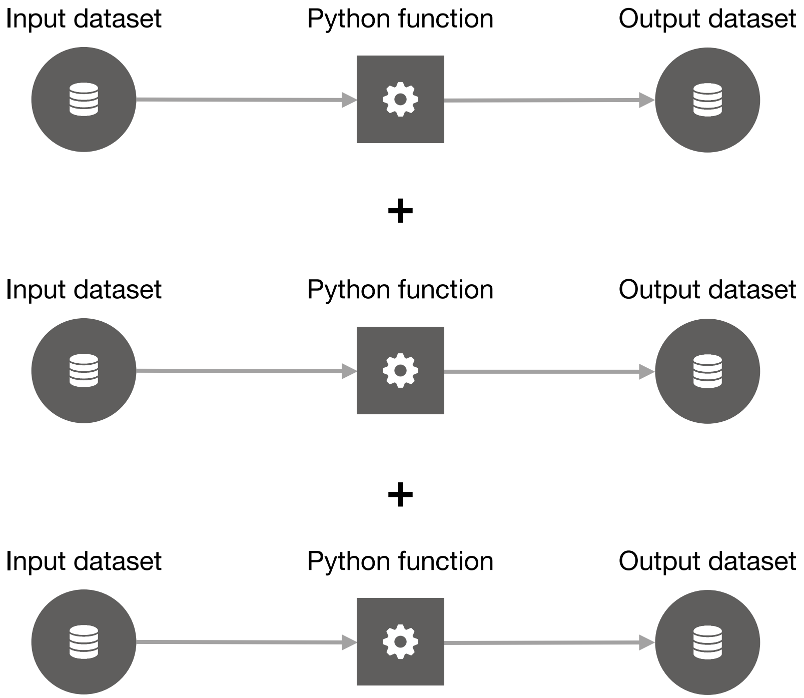
What are datasets?

- Usually an impure Python function that allows reading and writing to disk or other storage.
- All datasets are contained in the catalog and are accessible when defining a node.

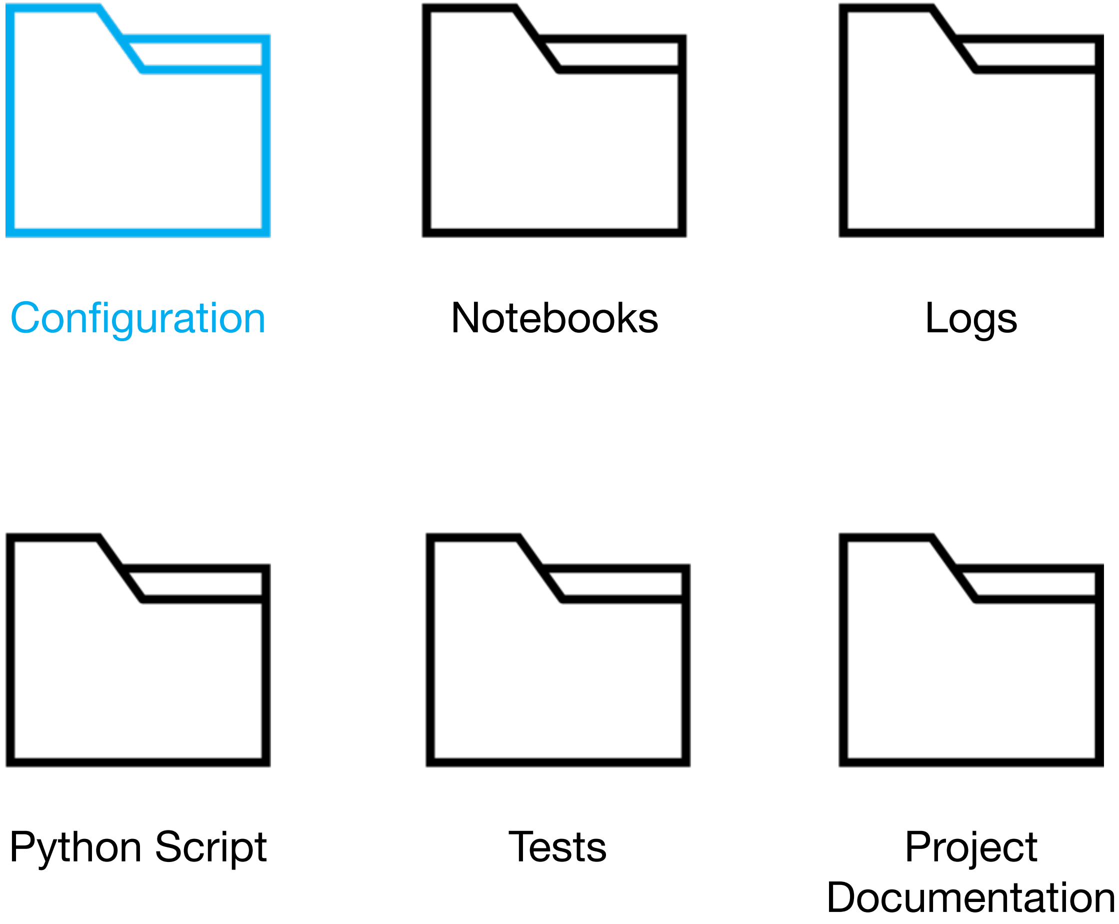


What is a pipeline?

- It is a directed acyclic graph.
- A collection of nodes with defined relationships and dependencies.



Configuration (Standardized Inputs and Outputs)



Configuration

Notebooks

Logs

Python Script

Tests

Project Documentation

What is configuration?

❖ “Settings” for your machine-learning code

❖ A way to define requirements for data, logging and parameters in different environments

❖ Helps keep credentials out of your code base

❖ Keep all parameters in one place

What does configuration help you do?

✓ Machine learning code that transitions from prototype to production with little effort

✓ Makes it possible to write generalizable and reusable analytics code that does not require significant modification to be used

Project Template (Convention for Organization)

Configuration

Notebooks

Logs

Python Script

Tests

Project Documentation

What is the project template?

❖ A modifiable series of files and folders

❖ Built-in support for Python logging, Pytest for unit tests and Sphinx for documentation

What does the project template help you do?

✓ Spend time on documenting your ML approach and not how your project is structured

✓ You spend less time digging around in previous projects for useful code

✓ Make it easier for collaborators to work with you

What are the benefits of Kedro?

Kedro continues to support a seamless experimentation to production workflow and has been fundamental in our ability to build reusable analytics code stores.

✓ **Consistent time to production**

Our teams can more accurately estimate the time required to produce production-ready code. There is also less time spent on refactoring and more time spent solving the business problem.

✓ **Reusable analytics code stores**

Kedro helps produce environment- and data- agnostic ML code, making code reusable. We are now benefiting from reusable code stores, significantly reducing time on use cases.

✓ **Increased collaboration**

Data engineers, data scientists, machine learning engineers and DevOps gain significant collaboration benefits because of the software engineering best-practice applied to the ML code base.

✓ **Upskilled developers**

Our users are learning about software engineering principles applied to ML code while they use Kedro and becoming more aware of best-practice when producing production-ready code.

INSTALL

```
> pip install kedro
```

Demo

Pipeline Visualisation

It gives you x-ray vision into your project. You can see exactly how data flows through your data and ML pipeline. It is fully automated and based on your code base.

PLUGIN

Demo:[*quantumblacklabs.github.io/kedro-viz/*](quantumblacklabs.github.io/kedro-viz/)

Flexible Deployment

Kedro supports packaging as a Python .egg or .whl. You can also produce documentation for your work. And choose to use deployment plugins for Docker and Airflow.

PLUGIN

What is Kedro-Docker?

- Kedro-Docker is a Kedro plugin, packages Kedro projects in Docker containers.
- This allows you to deploy Kedro code without worry about an operating system and installing dependencies
- This deployment mode facilitates action or time-triggered pipelines

Deployment Strategies with Kedro-Docker

- Use Kedro, Kedro-Docker and Kubernetes
- You can take advantage of Kubernetes abilities to orchestrate containers

PLUGIN

What is Kedro-Airflow?

- Kedro-Airflow, a Kedro plugin, converts Kedro pipelines into Airflow DAGs
- Kedro is much easier to setup and use than Airflow
- However, with Airflow you can take advantage of monitoring, scheduling and orchestrating functionality
- With Kedro-Airflow it’s easy to prototype your pipeline before deploying it

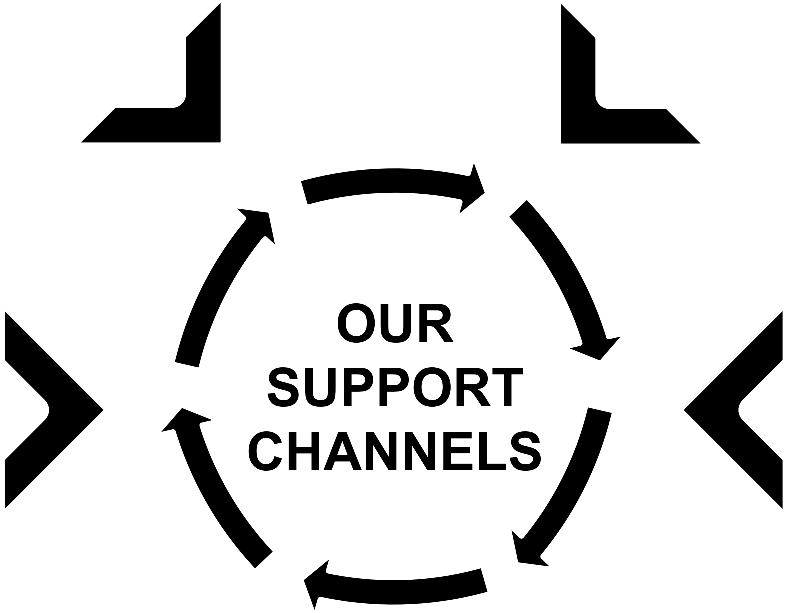
Kedro is actively maintained by QuantumBlack

We are committed to growing community and making sure that our users are supported for their standard and advanced use cases.

Join our open slack channel #kedro-users to ask questions or learn more about kedro.

Questions tagged with kedro are watched on Stack Overflow.

Documentation is available on Kedro's Read The Docs: https://kedro.readthedocs.io/

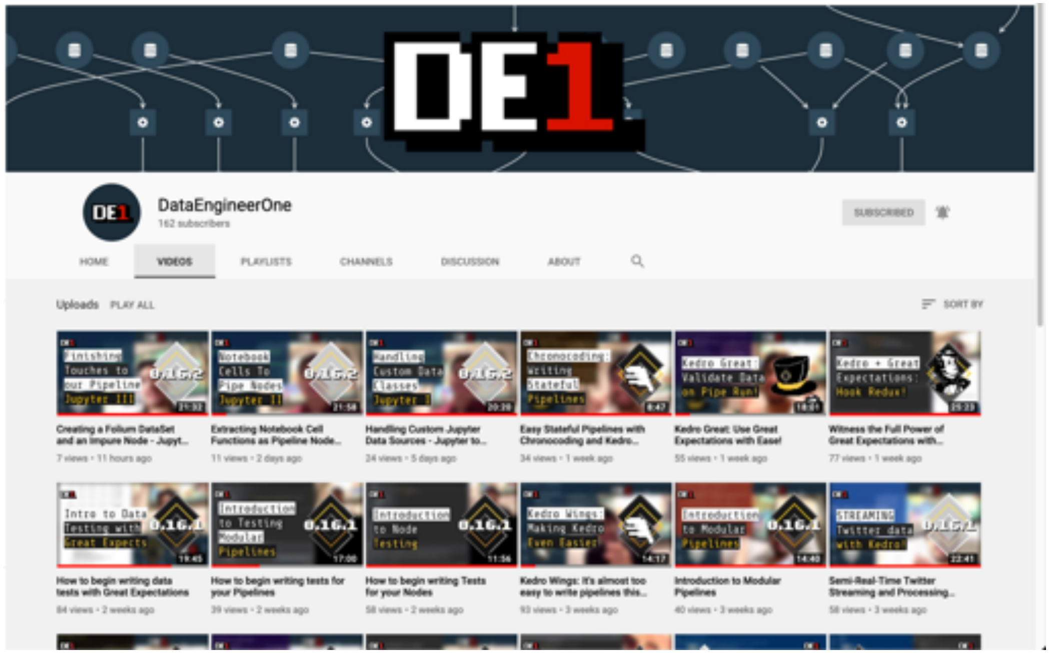


OUR SUPPORT CHANNELS

The Kedro community is active on: https://github.com/quantumblacklabs/kedro/
The team and contributors actively maintain raised feature requests, bug reports and pull requests.

KEDRO COMMUNITY

Kedro on YouTube



DataEngineerOne
162 subscribers

Creating a Folium DataSet and an Impure Node - Jupyt...
7 views • 11 hours ago

Extracting Notebook Cell Functions as Pipeline Node...
11 views • 2 days ago

Handling Custom Jupyter Data Sources - Jupyter to...
24 views • 5 days ago

Easy Stateful Pipelines with Chronocoding and Kedro...
34 views • 1 week ago

Kedro Great: Use Great Expectations with Ease!
55 views • 1 week ago

Witness the Full Power of Great Expectations with...
77 views • 1 week ago

How to begin writing data tests with Great Expectations
84 views • 2 weeks ago

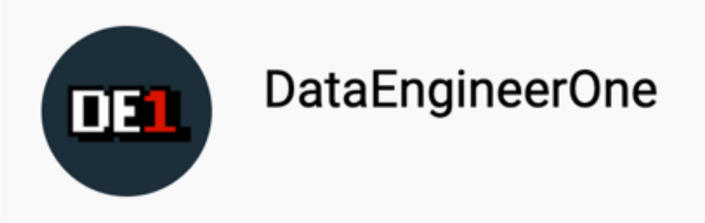
How to begin writing tests for your Pipelines
39 views • 2 weeks ago

How to begin writing Tests for your Nodes
58 views • 2 weeks ago

Kedro Wings: It's almost too easy to write pipelines this...
93 views • 3 weeks ago

Introduction to Modular Pipelines
40 views • 3 weeks ago

Semi-Real-Time Twitter Streaming and Processing...
58 views • 3 weeks ago



http://youtube.com/DataEngineerOne

THANK YOU!
QUESTIONS?

WHAT IS KEDRO?

The bridge between Machine Learning and Software Engineering

Kedro is an open source Python library, maintained by QuantumBlack. It is a development workflow tool that helps teams build data pipelines that are consistent, reproducible, versioned, scalable and deployable.

PROJECT TEMPLATE

A series of files and folders derived from Cookiecutter Data Science. Project setup consistency makes it easier for team members to collaborate with each other.

CONFIGURATION

Remove hard-coded variables from ML code so that it runs locally, in cloud or in production without major changes. Applies to data, parameters, credentials and logging.

THE CATALOG

An extensible collection of data, model or image connectors, available with a YAML or Code API, that borrow arguments from Pandas, Spark API and more.

NODES & PIPELINES

A pure Python function that has an input and an output. A pipeline is a directed acyclic graph, it is a collection of nodes with defined relationships and dependencies.

Kedro

USERS

Data Scientists
Data Engineers
Machine Learning Engineers

MATURITY

GROWTH