



Notebooks got your modern data *back*

Elizabeth Doha | Deepnote
Data Council Austin 2022



Agenda

01 Data science notebooks are *purple*



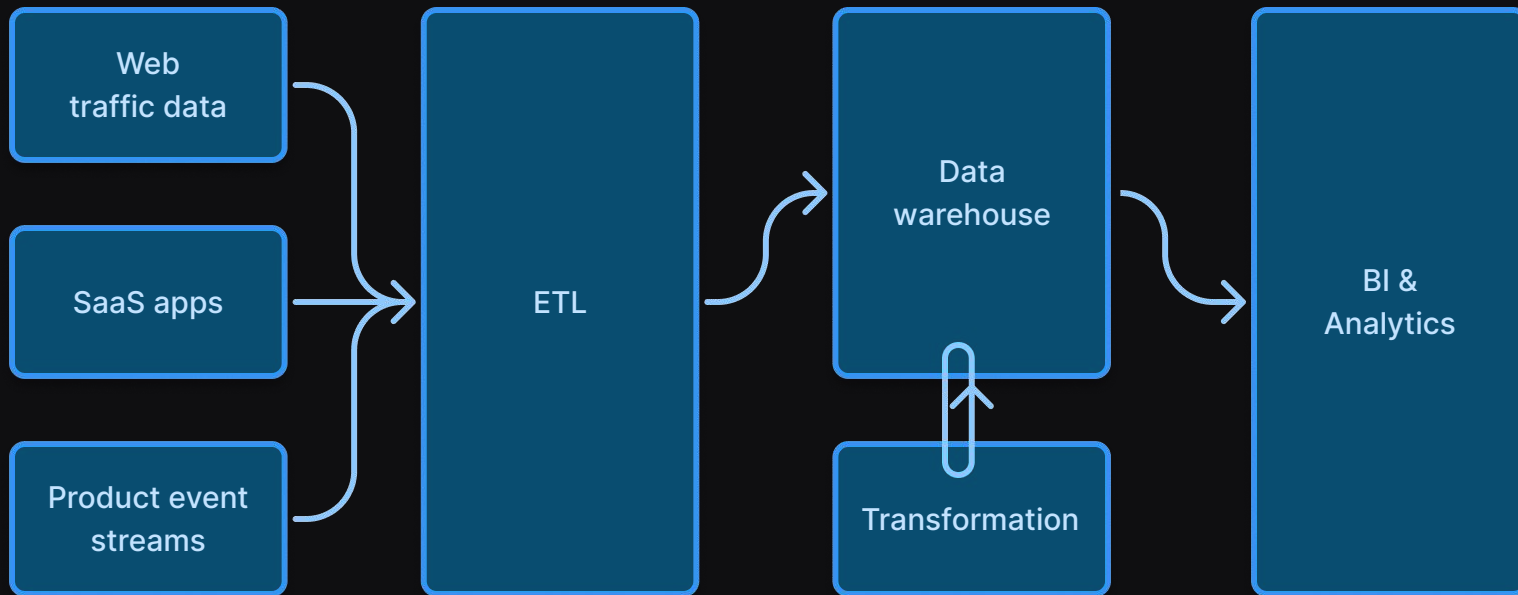
Agenda

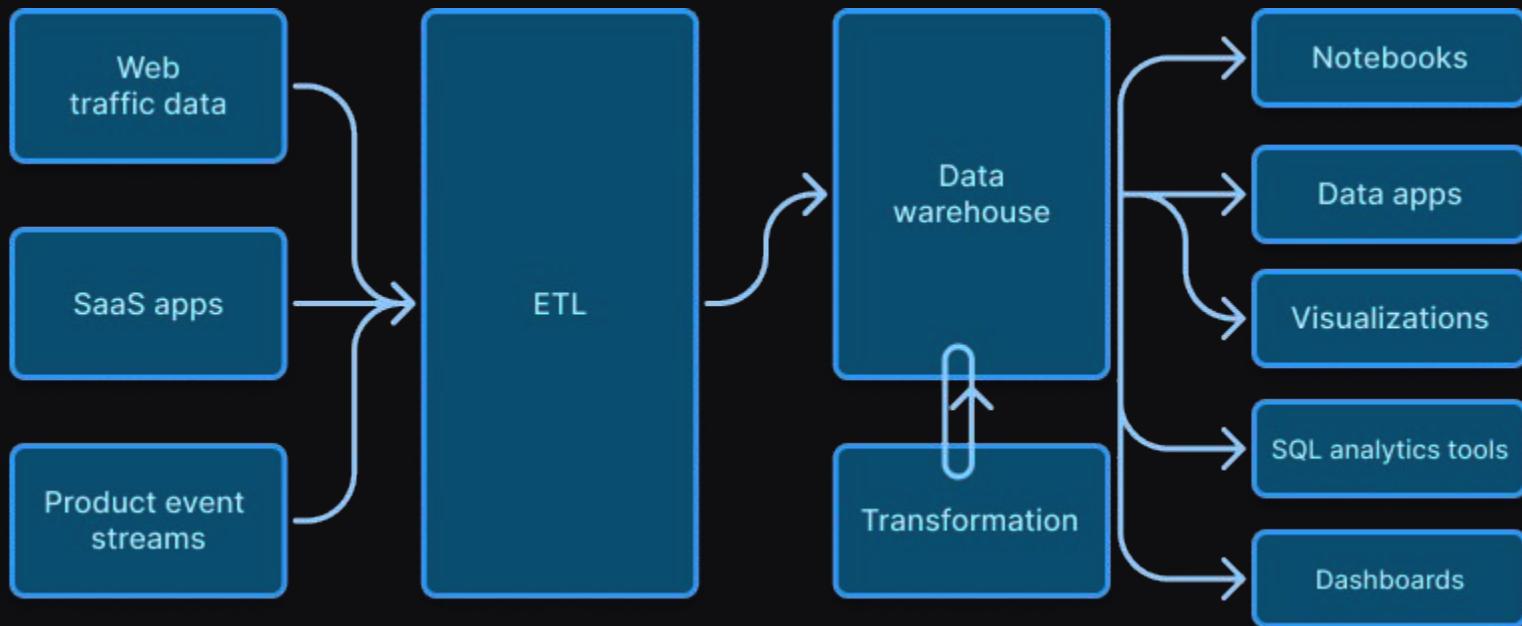
- 01 Data science notebooks are *purple*
- 02 Best practices for working in notebooks

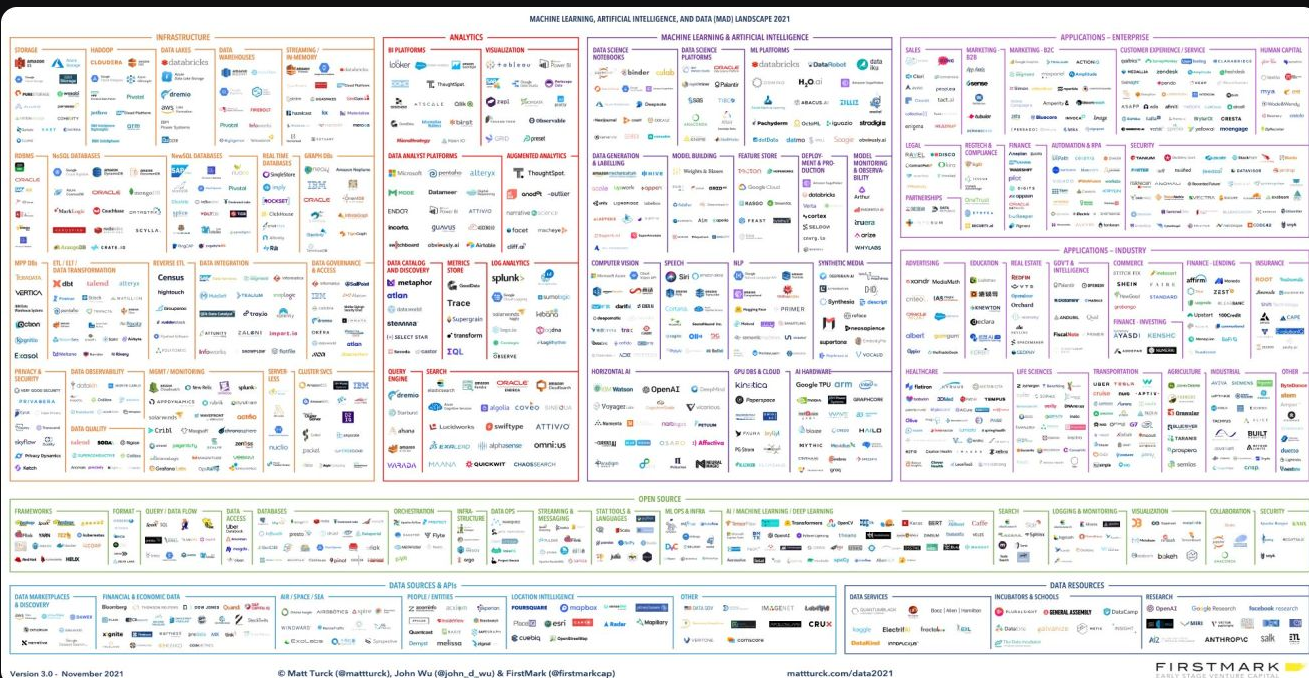


Agenda

- 01 Data science notebooks are *purple*
- 02 Best practices for working in notebooks
- 03 👁️ under the hood of Deepnote







SOURCE: MAD LANDSCAPE BY MATT TURCK





Data engineer



Business analyst



Analytics engineer



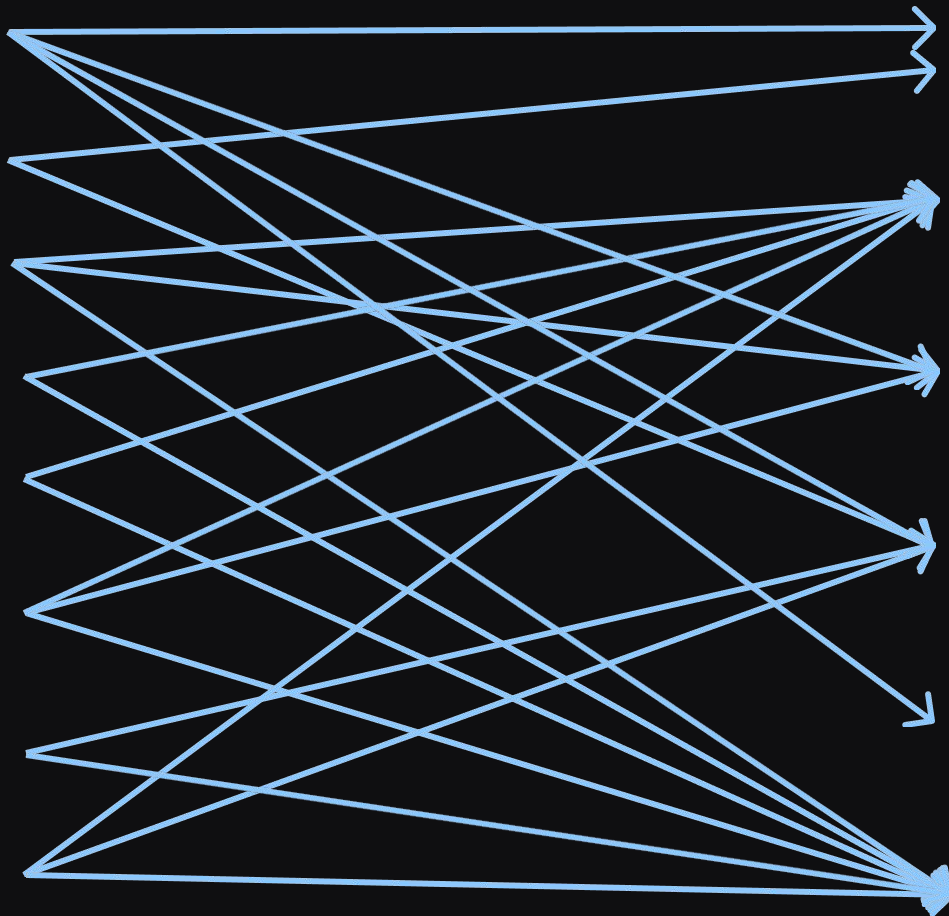
Data scientist









ML engineer



....



-  Data engineer
-  Business analyst
-  Analytics engineer
-  Data scientist
-  ML engineer
- 



1. How can we *retain & share* organizational knowledge?



SOURCE: [WE THE PURPLE PEOPLE](#) BY ANNA FILIPPOVA



💙 = Tech people (data producers)

❤️ = Business people (data consumers)





Engineering

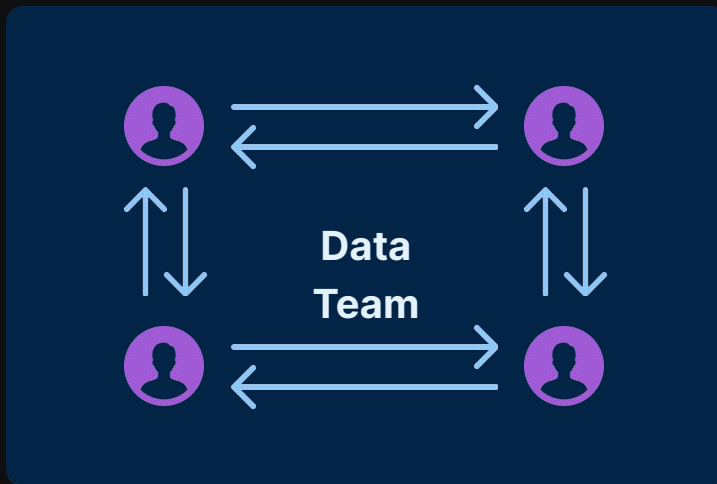


Data
Engineer

Product



Product
analyst



Marketing



Marketing
analyst

Finance



Data
scientist

Execs





Engineering



Data
Engineer

Marketing



Marketing
analyst

Finance



Data
scientist

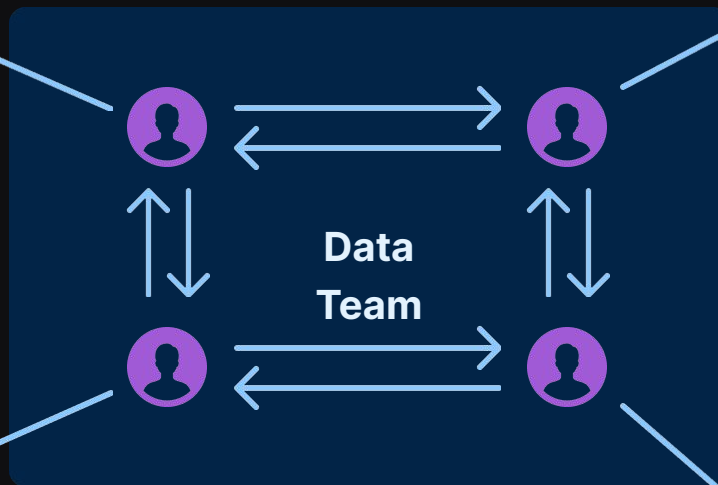
Execs



Product



Product
analyst





Engineering



Data
Engineer

Product



Product
analyst

Marketing



Marketing
analyst

Finance



Data
scientist

Execs





**Notebooks can be the *purple interface*
for the *purple people***



Notebooks 1.0

- ✓ General purpose
- ✓ Explainable,
transparent
- ✓ Shared context all in one place



2. How can we use notebooks *collaboratively*?



01

Follow established software development best practices

- Follow style guides, write documentation, write tests



01 Follow established software development best practices

- Follow style guides, write documentation, write tests

02 When training models, log all experiments automatically



01 Follow established software development best practices

- Follow style guides, write documentation, write tests

02 When training models, log all experiments automatically

03 Split development and production environments

- Sanitize user data in development environment



01 Follow established software development best practices

- Follow style guides, write documentation, write tests

02 When training models, log all experiments automatically

03 Split development and production environments

- Sanitize user data in development environment

04 Parametrize your notebooks



01 Follow established software development best practices

- Follow style guides, write documentation, write tests

02 When training models, log all experiments automatically

03 Split development and production environments

- Sanitize user data in development environment

04 Parametrize your notebooks

05 Continuous Integration (CI)



3. How do we make notebooks more *purple*?





Start with the *blue*



Language interoperability

COMBINE PYTHON AND SQL

station_name

BUFFALO NIAGARA INTERNATIONAL

✓

snowflake_demo

Saved to variable df_1


[10]

```
SELECT date, name, temp as mean_temp, wdsp as wind_speed, rain_drizzle as rain
FROM demo_db.public.ny_weather
WHERE date > '2015-12-31' and country = 'US' and name = {{(station_name)}}
ORDER BY date DESC
LIMIT 50
```

Preview

Visualize

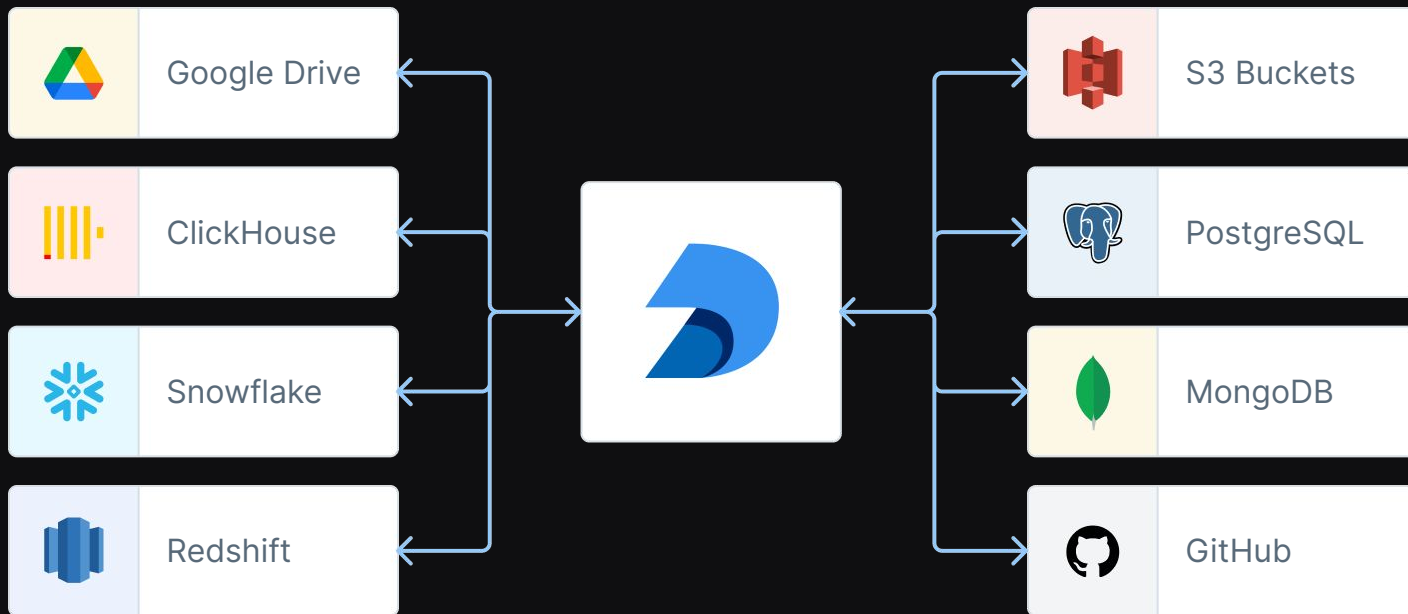
	date object	name object	mean_temp float64	wind_speed object	rain object
	2020-08-... .. 2%	BUFFALO NIAGARA IN... .. 100%	64.2 - 82.0	5.4 8%	0 100%
	2020-08-... .. 2%			4.3 4%	
	48 others 96%			34 others 88%	



0	2020-08-18	BUFFALO NIAGARA INTERNATIONAL	64.6	4.3	0
1	2020-08-17	BUFFALO NIAGARA INTERNATIONAL	71.3	5.4	0



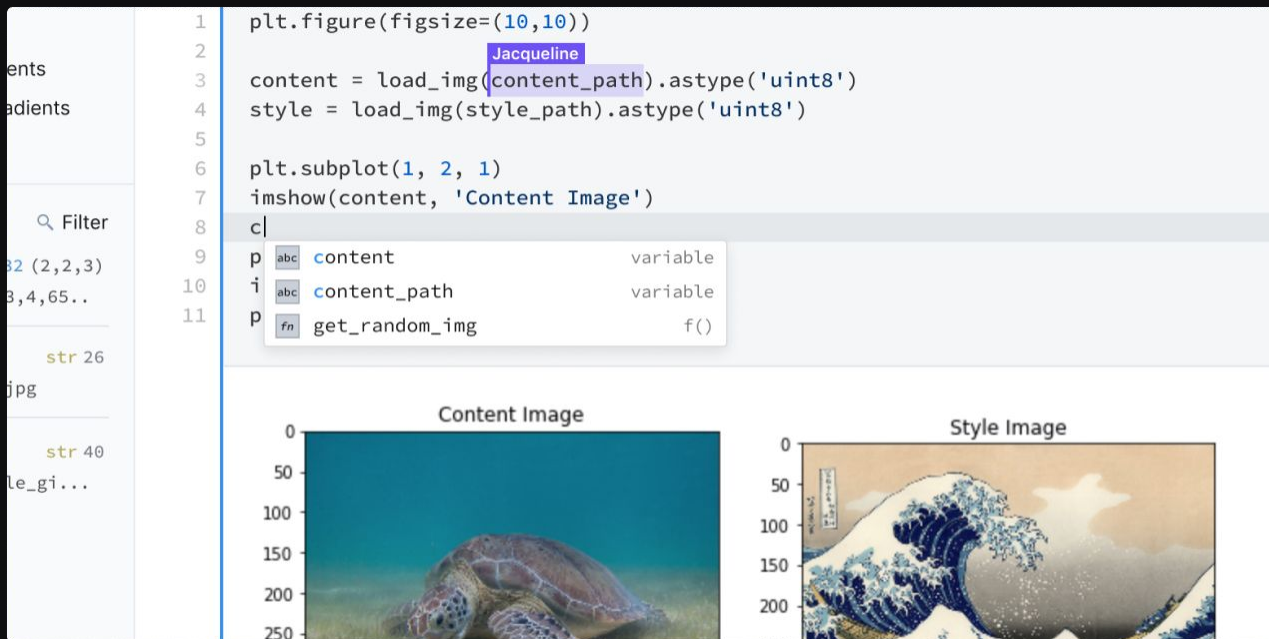
Integrations





Code intelligence

```
1 plt.figure(figsize=(10,10))
2
3 content = load_img(Jacqueline, content_path).astype('uint8')
4 style = load_img(style_path).astype('uint8')
5
6 plt.subplot(1, 2, 1)
7 imshow(content, 'Content Image')
8 c|
9 p abc content variable
10 i abc content_path variable
11 p fn get_random_img f()
```



The screenshot displays a Jupyter Notebook interface. The main area shows a code cell with the following Python code:

```
1 plt.figure(figsize=(10,10))
2
3 content = load_img(Jacqueline, content_path).astype('uint8')
4 style = load_img(style_path).astype('uint8')
5
6 plt.subplot(1, 2, 1)
7 imshow(content, 'Content Image')
8 c|
9 p abc content variable
10 i abc content_path variable
11 p fn get_random_img f()
```

The code completion dropdown at line 8 suggests the following options:

- `content` (variable)
- `content_path` (variable)
- `get_random_img` (function `f()`)

Below the code, two subplots are displayed side-by-side:

- Content Image:** A photograph of a sea turtle swimming underwater.
- Style Image:** A traditional Japanese woodblock print titled 'The Great Wave off Kanagawa' by Katsushika Hokusai.

The left sidebar shows a file explorer with a search bar labeled 'Filter'. Below it, there are several entries: '2 (2,2,3)', '3,4,65...', 'str 26', 'jpg', 'str 40', and 'le_gi...'.



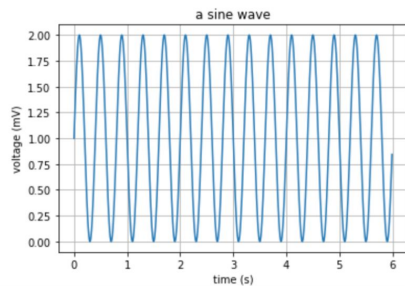
Mix in the *red*



Reactivity

```
# edit this value and see the notebook update live  
frequency = 5
```

```
t = np.arange(0.0, 6.0, 0.01)  
s = 1 + np.sin(frequency * np.pi * t)  
  
fig, ax = plt.subplots()  
ax.plot(t, s)  
  
ax.set(xlabel='time (s)', ylabel='voltage (mV)', title='a sine wave')  
ax.grid()  
  
plt.show()
```





Seamless editing (WYSIWYG)

The new editing experience

Gastropub chillwave umami lyft. Poke austin direct trade, marfa raclette letterpress actually. Chartreuse sriracha pinterest twee lo-fi try-hard. Meditation banh mi kitsch, prism organic hot chicken literally heirloom occupy af semiotics food truck.

Aesthetic asymmetrical gluten-free, health goth shaman meh bespoke kinfolk helvetica vaporware fashion axe freegan. Pour-over hammock succulents disrupt chartreuse raw denim. Brunch aesthetic

B

i

U

☒

😊

≡▼

≡▼

≡▼

≡

≡

🖼️

🔗

📄

↶

↷

pickled scenester venmo hashtag lo-fi.

ccie plaid pork

p. VHS blog



Data apps

SEC Annual Reports

Look for annual revenue for a company in a given fiscal year.

Companies pick different names for the measure to file the number, so we look for all measures containing the word "revenue".

company_name

NVIDIA CORP

Apply

Data Warehouse

Saved to variable filter_annual_report

```
SELECT
  company_name,
  fiscal_year,
  measure_tag,
FROM
  bigquery-public-data.sec_quarterly_financials.quick_summary
WHERE
  company_name = {(company_name)}
  AND LOWER(measure_tag) LIKE '%revenue%'
LIMIT 10
```

Visualize

	company_name o...	fiscal_year in...	measure_tag ob...
	NVIDIA CORP - 100%	2012 - 2012	NumberOfCu... 20% Revenues 20% 6 others 60%
0	NVIDIA CORP	2012	NumberOfCustome... rsWithSignific...
1	NVIDIA CORP	2012	DeferredRevenue Noncurrent
2	NVIDIA CORP	2012	Revenues
3	NVIDIA CORP	2012	Revenues
4	NVIDIA CORP	2012	CostOfRevenue

SEC Annual Reports

Look for annual revenue for a company in a given fiscal year.

Companies pick different names for the measure to file the number, so we look for all measures containing the word "revenue".

company_name

NVIDIA CORP

SQL

Saved to variable filter_annual_report

	company_name o...	fiscal_year int...	measure_tag obj...
	NVIDIA CORP - 100%	2012 - 2012	NumberOfCu... 20% Revenues 20% 6 others 60%
0	NVIDIA CORP	2012	NumberOfCustome... rsWithSignific...
1	NVIDIA CORP	2012	DeferredRevenue Noncurrent
2	NVIDIA CORP	2012	Revenues
3	NVIDIA CORP	2012	Revenues
4	NVIDIA CORP	2012	CostOfRevenue
5	NVIDIA CORP	2012	NumberOfCustome... rsWithSignific...
6	NVIDIA CORP	2012	AllocatedShareB... asedCompensati...
7	NVIDIA CORP	2012	NetWarrantyChar... geAgainstCostO...
8	NVIDIA CORP	2012	DeferredRevenue Current
9	NVIDIA CORP	2012	RevenueFromSign... ificantCustome...



Dive into *purple*



Collaboration

REAL-TIME COLLABORATION

Deepnote (Filip)

Find a random example and show its corresponding label

Jacqueline

```
rand_example = np.random.choice(training_examples)
_, ax = plt.subplots(1, 1)
ax.imshow(training_examples[rand_example])
ax.set_title("Label: %i" % training_targets.loc[rand_example])
ax.grid(False)
```

5.4s Executing cell


0 5 10 15 20 25

0

5

10

15



Deepnote (Jacqueline)

Find a random example and show its corresponding label

```
rand_example = np.random.choice(training_examples)
_, ax = plt.subplots(1, 1)
ax.imshow(training_examples[rand_example].values)
ax.set_title("Label: %i" % training_targets.loc[rand_example])
ax.grid(False)
```

5.4s Executing cell



Collaboration

COMMENTS

```
stocks=pd.read_csv('stocks.csv')
stocks
```

✓

	date	object	AAPL	float64	AMZN	float64	GOOG	float64			
	2000-01-01	...	0.8%	7.07	-	223.02	5.97	-	135.91	102.37	-
	2000-02-01	...	0.8%								
	121	others	98.4%								

0	2000-01-01	25.94	64.56	nan
1	2000-02-01	28.66	68.87	nan
2	2000-03-01	33.95	67	nan
3	2000-04-01	31.01	55.19	nan

Show 2 resolved threads

Allan Campopiano (You) 5 months ago

Can you transform this data into long form? I'll start writing the code for a chart.

Sam Salvatore 5 months ago

Absolutely I can do that!

+ Add comment

⌘ + ⌘ + C

✓ Resolve



Knowledge organization

FOLDERS & SHARING

- Workspace +
 - > Admin
 - > Analytics
 - > Lead scoring
 - Advent 2018
 - Churn prediction
 - Credit card fraud prediction...
 - Exploring BigQuery SEC da...
 - Retail sales forecast
 - > Knowledge Base
 - > New in Deepnote
 - > Production
 - Retarget users about to ch...

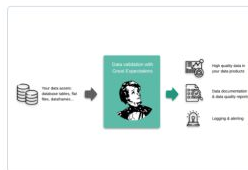
TEMPLATES

Create new project from template



Daily Stock Price Dashboard

Analyze historical stock market data from Yahoo! finance



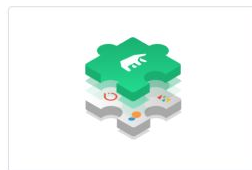
Reduce Pipeline Debt With Great Expectations

Use Great Expectations to build tests and validate columns in your data



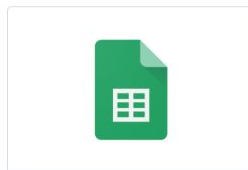
A/B Testing

Use statistics to test for differences in conversion and retention rates



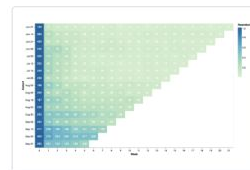
Machine Learning With SQL

Use MindsDB to predict customer satisfaction with SQL



Read Google Sheets

Read your Google Sheets into a Pandas DataFrame



User Retention Charts

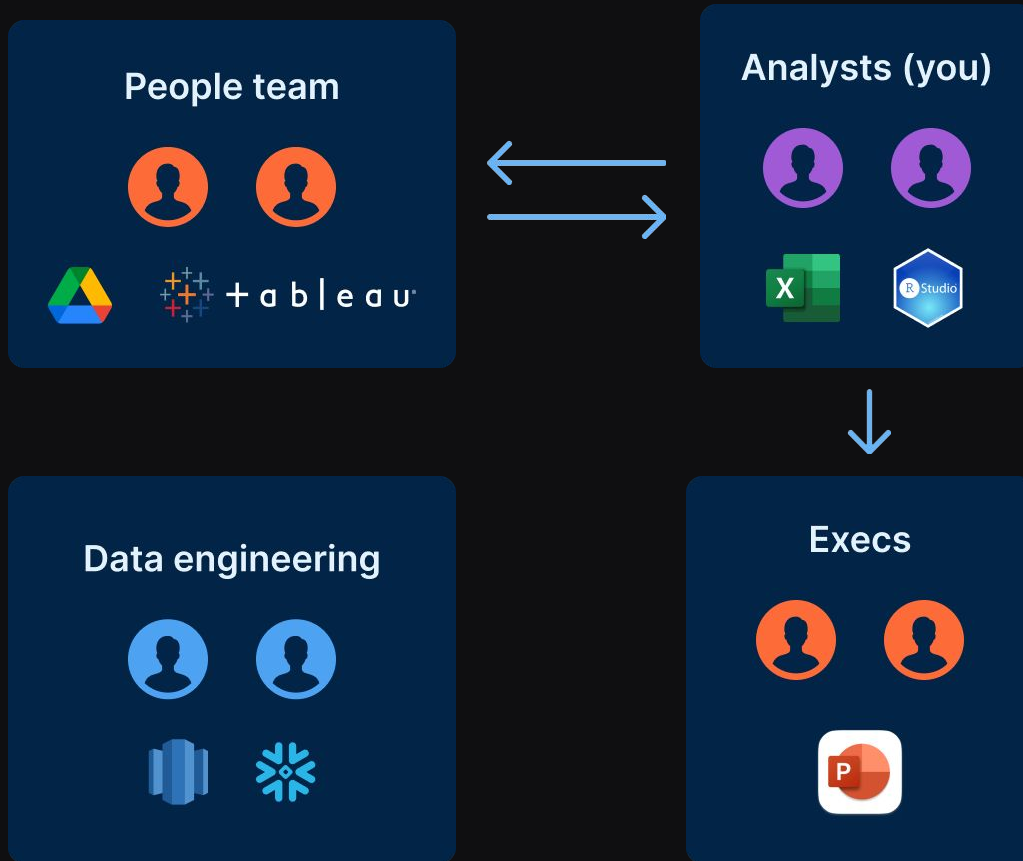
Visualize cohorts of users over time to measure retention






4. What does this *purple* world look like?

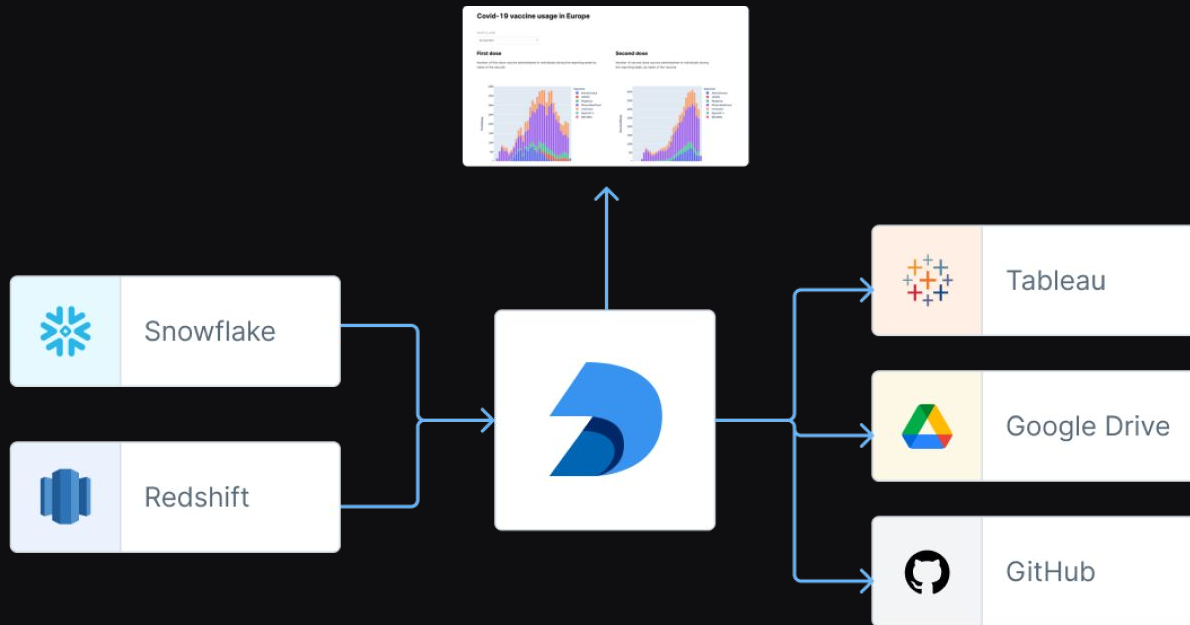
Before




After


Data
engineering


You




People team


Execs

Outcomes

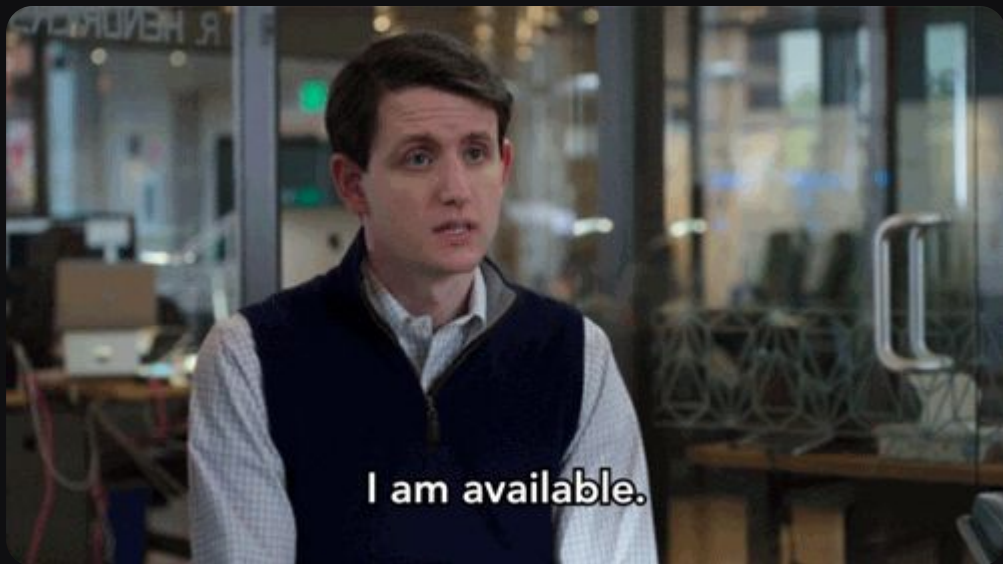
- ✓ Streamlined workflows
- ✓ Reduced time-to-insight by breaking down silos
- ✓ Hues of purple everywhere



Questions?

Elizabeth Doha

elizabeth@deepnote.com





FAREWELL DRINKS & GAMES

With Deepnote & dbt Labs



Deepnote



dbt Labs

Case study: How Gusto scales their data capabilities

You're an analyst on the People team, helping your company make data-driven decisions about workforce.

Questions such as...

1. What are the key drivers of attrition?
2. How does remote impact employee well-being?





Versioning

Let's create methods that will allow us to load and preprocess images. The preprocessing process as are expected according to the model trained on image with each channel normalized by the mean of the channels BGR.

```
def load_and_process_img(path_to_img):  
    img = load_img(path_to_img)  
    img = tf.keras.applications.vgg19.preprocess_image(img)  
    return image
```

In order to view to look at the outputs of our optimization step. Furthermore, since our optimized image may take values outside the 0-255 range, we must clip to maintain our values from within the 0-255 range.


```
def deprocess_img(processed_img):
```

- ● Peter Taylor
May 2 3:03pm [Current Version](#)
- ● Núria Moura
May 2 11:49 am [8 changes](#)
- ● Peter Taylor
May 1 4:55pm [11 changes](#)
- ● Filip Jakobsson
April 30 2:31pm [2 changes](#)
- ● Jacqueline Asong
April 25 2:43pm [27 changes](#)
- ● Núria Moura
April 25 1:34pm [9 changes](#)



Productionize instantly

SCHEDULING

 Schedule notebook

×

Runs the notebook on a daily or weekly schedule. You can see the previous runs in the [history tab](#).

Schedule ☒

Daily

at

09:00 AM

Europe/Prague

▼ Advanced settings

When a notebook runs successfully...

☒ Notify me via email

Send email notification when a run succeeds.

☒ Re-publish notebook

When a scheduled run is completed, the notebook will be republished. This works only if the project is currently published.

When a notebook run results in an error...

☒ Notify me via email

Send email notification when a run fails.

Save schedule