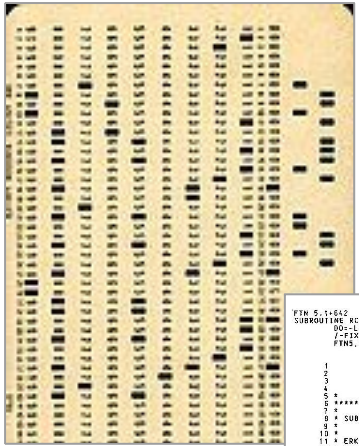
Einblick

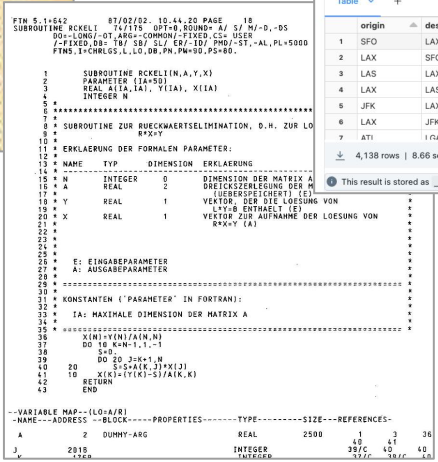
Product Overview for Databricks

AI will create 7 billion new data scientists

Historically, data analysis was a complex technical endeavor. Today, AI empowers everyone to find the right answers simply by asking natural language questions.



1940



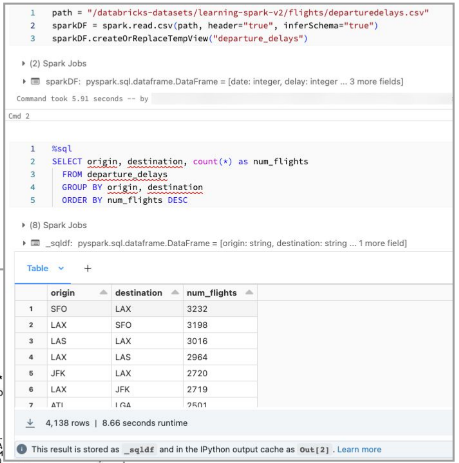
```
FTN 5.1+642        87/02/02. 10.44.20 PAGE    18
SUBROUTINE RCKELI    74/175  OPT=0,ROUND= A/ S/ M/-D,-DS
        DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
        /-FIXED,DB= TB/ SB/ SL/ ER/-ID/ PMD/-ST,-AL,PL=5000
        FTN5,I=CHRLGS,L,LO,DB,PN,PW=90,PS=80.

          SUBROUTINE RCKELI(N,A,Y,X)
          PARAMETER (IA=50)
          REAL A(IA,IA), Y(IA), X(IA)
          INTEGER N
      *
      ***************************************************
      *
      *  SUBROUTINE ZUR RUECKWAERTSELIMINATION, D.H. ZUR LO
      *                R*X=Y
      *
      *  ERKLAERUNG DER FORMALEN PARAMETER:
      *
      *  NAME    TYP      DIMENSION  ERKLAERUNG
      *  ----------------------------------------------------
      *  N       INTEGER     0       DIMENSION DER MATRIX A
      *  A       REAL        2       DREICKSZERLEGUNG DER M
      *                                (UEBERSPEICHERT) (E)
      *  Y       REAL        1       VEKTOR, DER DIE LOESUNG VON          *
      *                                L*Y=B ENTHAELT (E)                 *
      *  X       REAL        1       VEKTOR ZUR AUFNAHME DER LOESUNG VON  *
      *                                R*X=Y (A)                          *
      *                                                                   *
      *                                                                   *
      *                                                                   *
      *                                                                   *
      *    E: EINGABEPARAMETER                                            *
      *    A: AUSGABEPARAMETER                                            *
      *                                                                   *
      *  ================================================================ *
      *                                                                   *
      *  KONSTANTEN ('PARAMETER' IN FORTRAN):                             *
      *                                                                   *
      *    IA: MAXIMALE DIMENSION DER MATRIX A                            *
      *                                                                   *
      *  ================================================================ *
          X(N)=Y(N)/A(N,N)
          DO 10 K=N-1,1,-1
             S=0.
             DO 20 J=K+1,N
   20           S=S+A(K,J)*X(J)
   10        X(K)=(Y(K)-S)/A(K,K)
          RETURN
          END

--VARIABLE MAP--(LO=A/R)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-

  A          2   DUMMY-ARG                   REAL          2500       1      3     36
                                                                     40     41
 J        201B                              INTEGER                39/C    40     40
 K        176B                              INTEGER                37/C   39/C    40
```

1960



```
path = "/databricks-datasets/learning-spark-v2/flights/departuredelays.csv"
sparkDF = spark.read.csv(path, header="true", inferSchema="true")
sparkDF.createOrReplaceTempView("departure_delays")
```

▸ (2) Spark Jobs

▸ sparkDF: pyspark.sql.dataframe.DataFrame = [date: integer, delay: integer ... 3 more fields]

Command took 5.91 seconds -- by [redacted]

Cmd 2

```
%sql
SELECT origin, destination, count(*) as num_flights
  FROM departure_delays
  GROUP BY origin, destination
  ORDER BY num_flights DESC
```

▸ (8) Spark Jobs

▸ _sqldf: pyspark.sql.dataframe.DataFrame = [origin: string, destination: string ... 1 more field]

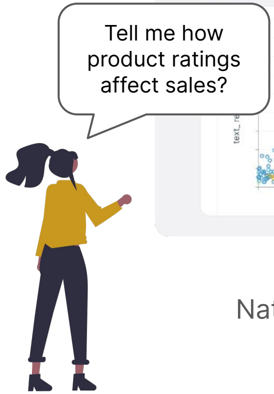
Table +

	origin	destination	num_flights
1	SFO	LAX	3232
2	LAX	SFO	3198
3	LAS	LAX	3016
4	LAX	LAS	2964
5	JFK	LAX	2720
6	LAX	JFK	2719
7	ATL	LGA	2501

4,138 rows | 8.66 seconds runtime

This result is stored as `_sqldf` and in the IPython output cache as `Out[2]`. Learn more

2020



log_ reviews_ count
log_ ratings_ count
text_ re
ratings_ count

Now

Natural language replaces programming

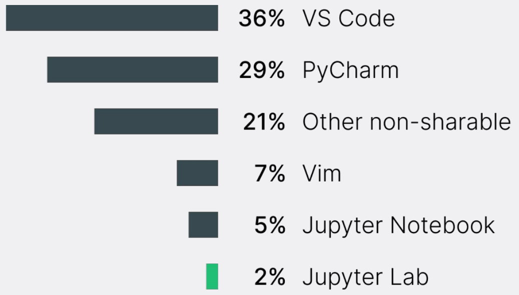
Einblick

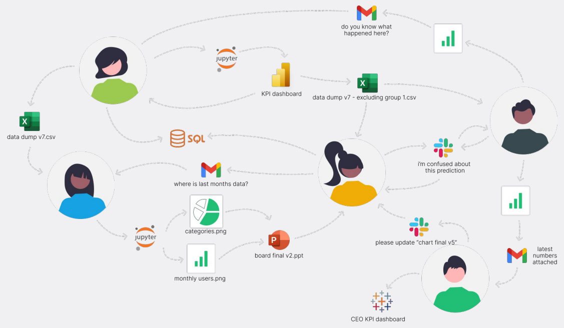
Current Trends

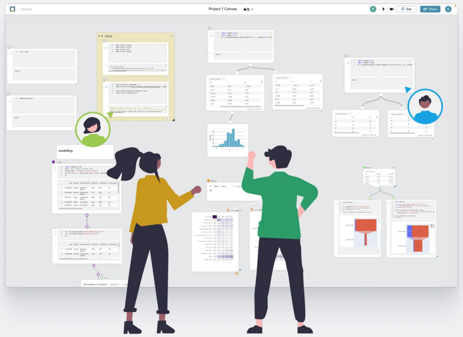
98% of Python is written in local editors

BI / analytics and data science are converging

Democratization of data analytics / science

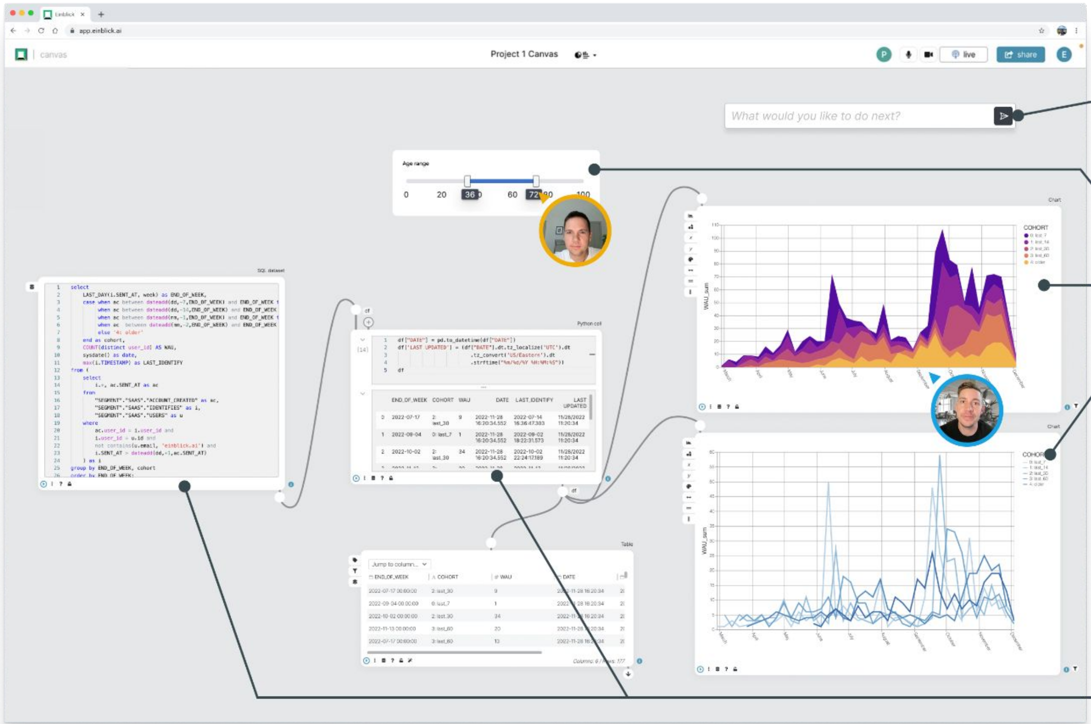



do you know what happened here?
jupyter
KPI dashboard
data dump v7 - excluding group 1.csv
data dump v7.csv
SQL
i'm confused about this prediction
where is last months data?
categories.png
jupyter
monthly users.png
board final v2.ppt
please update "chart final v5"
latest numbers attached
CEO KPI dashboard



What is Einblick?

A next-generation, AI-native, multi-modal data notebook to build workflows and data apps



```
select
    LAST_DAY(i.SENT_AT, week) as END_OF_WEEK,
    case when ac between dateadd(dd,-7,END_OF_WEEK) and END_OF_WEEK t
         when ac between dateadd(dd,-14,END_OF_WEEK) and END_OF_WEEK
         when ac between dateadd(mm,-1,END_OF_WEEK) and END_OF_WEEK t
         when ac  between dateadd(mm,-2,END_OF_WEEK) and END_OF_WEEK
         else '4: older'
    end as cohort,
    COUNT(distinct user_id) AS WAU,
    sysdate() as date,
    max(i.TIMESTAMP) as LAST_IDENTIFY
from (
    select
        i.*, ac.SENT_AT as ac
    from
        "SEGMENT"."SAAS"."ACCOUNT_CREATED" as ac,
        "SEGMENT"."SAAS"."IDENTIFIES" as i,
        "SEGMENT"."SAAS"."USERS" as u
    where
        ac.user_id = i.user_id and
        i.user_id = u.id and
        not contains(u.email, 'einblick.ai') and
        i.SENT_AT > dateadd(dd,+1,ac.SENT_AT)
    ) as i
group by END_OF_WEEK, cohort
order by END_OF_WEEK;
```

```
df["DATE"] = pd.to_datetime(df["DATE"])
df['LAST UPDATED'] = (df["DATE"].dt.tz_localize('UTC').dt
                              .tz_convert('US/Eastern').dt
                              .strftime("%m/%d/%Y %H:%M:%S"))
df
```

	END_OF_WEEK	COHORT	WAU	DATE	LAST_IDENTIFY	LAST UPDATED
0	2022-07-17	2: last_30	9	2022-11-28 16:20:34.552	2022-07-14 16:36:47.303	11/28/2022 11:20:34
1	2022-09-04	0: last_7	1	2022-11-28 16:20:34.552	2022-09-02 18:22:31.573	11/28/2022 11:20:34
2	2022-10-02	2: last_30	34	2022-11-28 16:20:34.552	2022-10-02 22:24:17.189	11/28/2022 11:20:34

END_OF_WEEK	COHORT	WAU	DATE
2022-07-17 00:00:00	2: last_30	9	2022-11-28 16:20:34
2022-09-04 00:00:00	0: last_7	1	2022-11-28 16:20:34
2022-10-02 00:00:00	2: last_30	34	2022-11-28 16:20:34
2022-11-13 00:00:00	3: last_60	20	2022-11-28 16:20:34
2022-07-17 00:00:00	3: last_60	13	2022-11-28 16:20:34

Natural language

Explore data, build workflows and refine cells by having a conversation with the system.

No-code

No-code operators for common tasks (visualizations, table manipulations, automated ML, etc.) to enhance productivity and empower users of all technical backgrounds. Input controls to build data apps.

Code

Native support for SQL and Python. Pull data from BiqQuery, Snowflake, etc. and pipe into Python cells which are functionally equivalent to Jupyter notebooks.

Target Persona

Editors

Data practitioners, folks that are familiar with Python and / or SQL

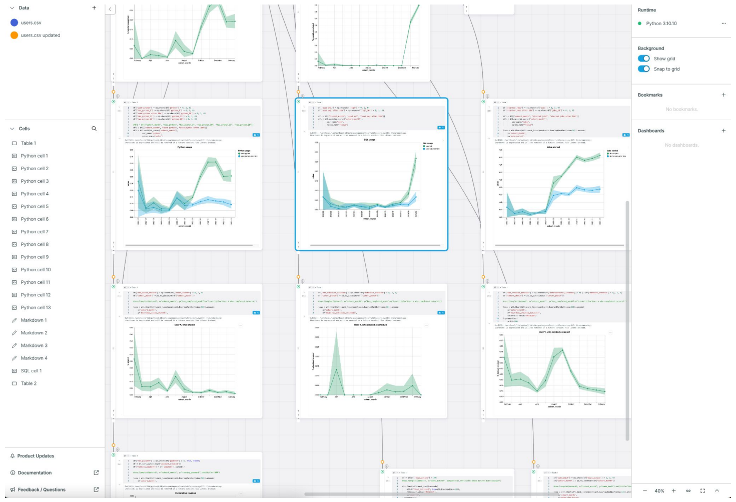
(still early, but clear indication that barrier to entry significantly lowered through AI)

Viewers

Citizen data scientists, stakeholders in the analytics process, oftentimes less technical

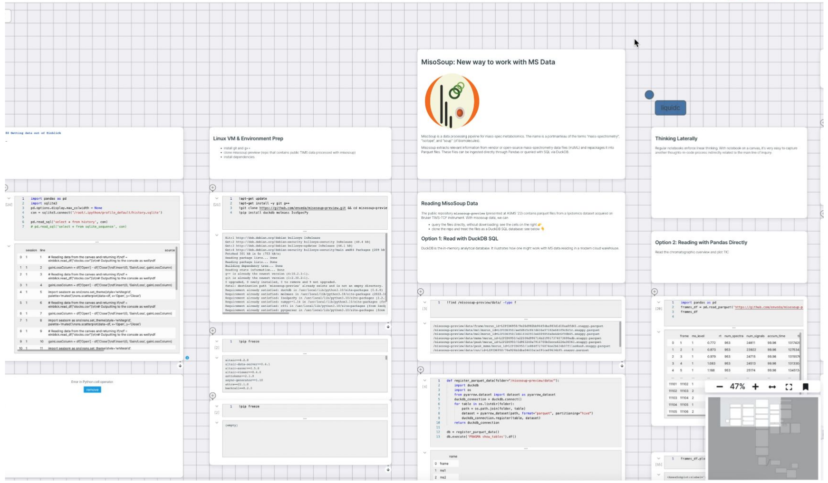
Use Cases

Advanced reporting



Data in SQL & wrangling in Python
Recurring

Data science experimentation


MisoSoup: New way to work with MS Data
Linux VM & Environment Prep
Reading MisoSoup Data
Option 1: Read with DuckDB SQL
Thinking Laterally
Option 2: Reading with Pandas Directly
47%

Python, ML, SQL, no-code
Ad-hoc

Who are we?

Making data analytics a more effortless, efficient, and collaborative experience.

Founded in 2020 as a remote first company based primarily in Boston and NYC with talent in applied machine learning and frontend engineering

6+ years of research at the intersection of human-computer interaction (HCI) and machine learning at MIT and Brown University.

Amplify Partners | DARPA | Dell | Flybridge Capital Partners | Intel Capital | Samsung Next

10M in total funding

Emanuel Zgraggen
CEO / Co-founder

- Ex-Postdoc @ MIT
- Ph.D. in Computer Science from Brown University
- Passion for products at the intersection of HCI and machine learning / AI
- Research[1] in HCI and applied ML, building interactive tools for data analysis and data science

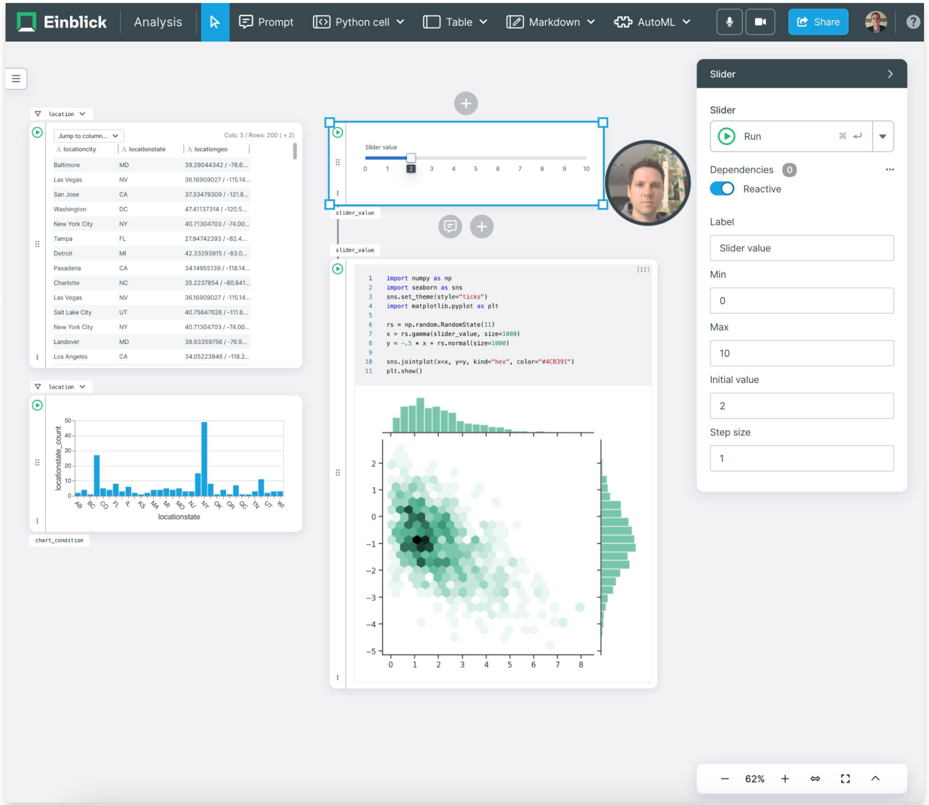
Philipp Eichmann
CXO / Co-founder

- Ph.D. in Computer Science from Brown University
- Life-long UI/UX enthusiast and engineer
- Passionate about designing and building tools for data science
- Conducted research[2] in HCI to democratize access to data science

[1] bit.ly/3vKvNO2
[2] bit.ly/3Oh0gcO

User Interface

User Interface



Cols: 3 / Rows: 200 (+ 2)

locationcity	locationstate	locationgeo
Baltimore	MD	39.29044342 / -76.6...
Las Vegas	NV	36.16909027 / -115.14...
San Jose	CA	37.33479309 / -121.8...
Washington	DC	47.41137314 / -120.5...
New York City	NY	40.71304703 / -74.00...
Tampa	FL	27.94742393 / -82.4...
Detroit	MI	42.33293915 / -83.0...
Pasadena	CA	34.14955139 / -118.14...
Charlotte	NC	35.2237854 / -80.841...
Las Vegas	NV	36.16909027 / -115.14...
Salt Lake City	UT	40.75847626 / -111.8...
New York City	NY	40.71304703 / -74.00...
Landover	MD	38.93359756 / -76.9...
Los Angeles	CA	34.05223846 / -118.2...

[11]

```python
import numpy as np
import seaborn as sns
sns.set_theme(style="ticks")
import matplotlib.pyplot as plt

rs = np.random.RandomState(11)
x = rs.gamma(slider_value, size=1000)
y = -.5 * x + rs.normal(size=1000)

sns.jointplot(x=x, y=y, kind="hex", color="#4CB391")
plt.show()
```

Slider

Slider

Run

Dependencies 0

Reactive

Label: Slider value

Min: 0

Max: 10

Initial value: 2

Step size: 1

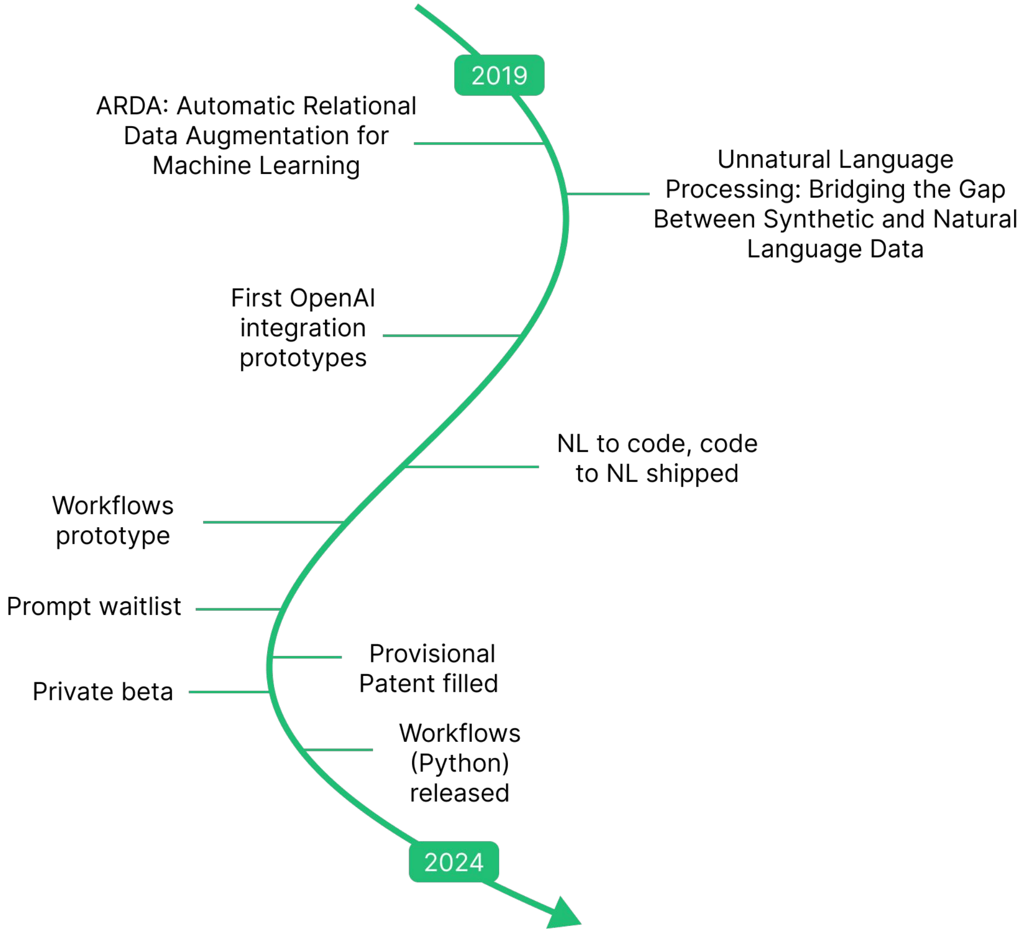
Problem	Solution
Notebooks preclude non-technical collaborators	Completely re-thought notebook/canvas design, library of no-code cells
Collaboration is crucial	Collaboration is a first principle: canvas as an emerging means for collaboration, real-time video/audio streaming
The possibilities for no-code operators are endless	Extensible architecture and plugin Infrastructure
BI and Data Science are converging	Multi-modal environment: no-code, code, and natural language
Notebooks cannot easily be turned interactive, shareable experiences	User controls and reactive execution, data apps
Interpretability and reproducibility of notebooks can be confusing and hard	Automatic dependency resolution using static code analysis, exposed to the user

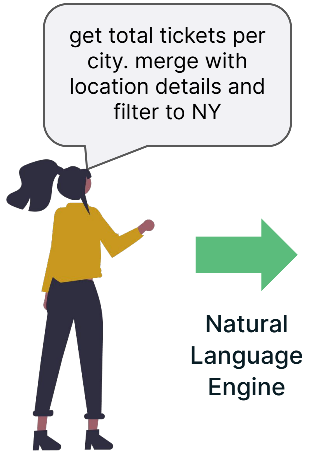
Natural Language Engine

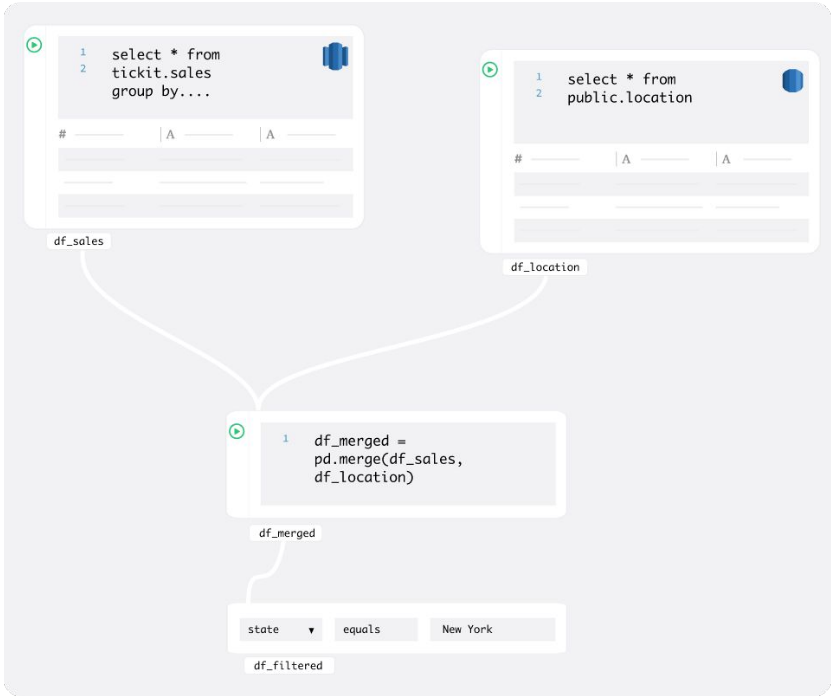
From text to data workflow

Why we care:

- Time saver for experts
- Lower barrier of entry for non-experts


2019
ARDA: Automatic Relational Data Augmentation for Machine Learning
Unnatural Language Processing: Bridging the Gap Between Synthetic and Natural Language Data
First OpenAI integration prototypes
NL to code, code to NL shipped
Workflows prototype
Prompt waitlist
Provisional Patent filled
Private beta
Workflows (Python) released
2024


get total tickets per city. merge with location details and filter to NY
Natural Language Engine



```
select * from
tickit.sales
group by....
```

df_sales

```
select * from
public.location
```

df_location

```
df_merged =
pd.merge(df_sales,
df_location)
```

df_merged

state ▾ equals New York

df_filtered

Targeted: best possible AI engine for the data analytics and data science

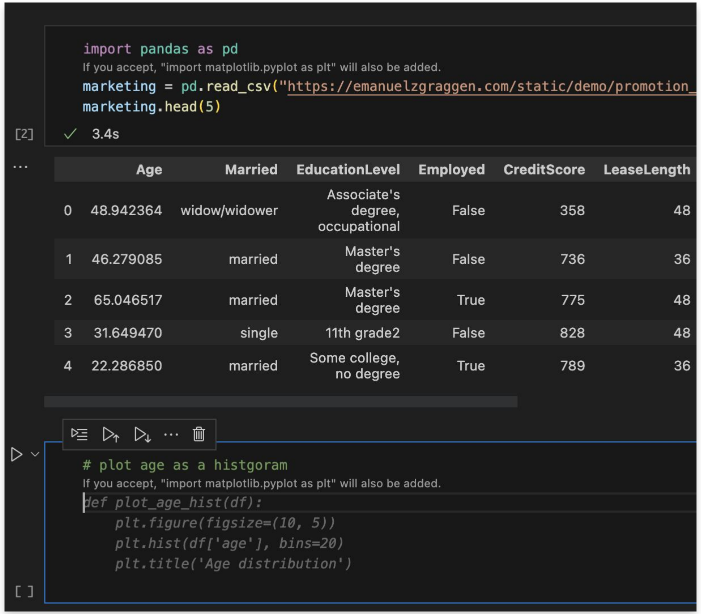
Multimodal: mix no-code, Python, SQL

Extensible: incorporate new task types and modalities

Adaptable: keep up to date with the most recent advancements

https://www.einblick.ai/blog/einblick-prompt-vs-github-copilot/
https://www.einblick.ai/blog/einblick-prompt-vs-jupyter-ai/

Example: “Plot age as a histogram”

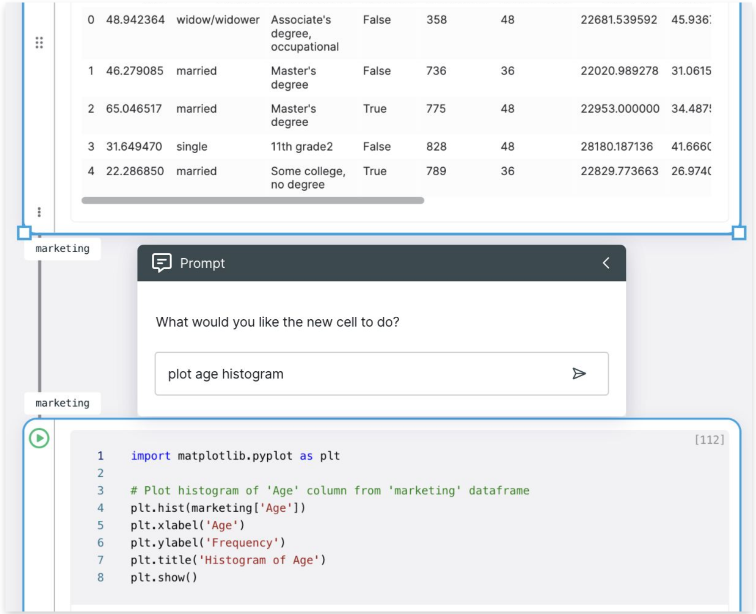


```
import pandas as pd
If you accept, "import matplotlib.pyplot as plt" will also be added.
marketing = pd.read_csv("https://emanuelzgraggen.com/static/demo/promotion_
marketing.head(5)
```

[2] ✓ 3.4s

	Age	Married	EducationLevel	Employed	CreditScore	LeaseLength
0	48.942364	widow/widower	Associate's degree, occupational	False	358	48
1	46.279085	married	Master's degree	False	736	36
2	65.046517	married	Master's degree	True	775	48
3	31.649470	single	11th grade2	False	828	48
4	22.286850	married	Some college, no degree	True	789	36

```
# plot age as a histgoram
If you accept, "import matplotlib.pyplot as plt" will also be added.
def plot_age_hist(df):
    plt.figure(figsize=(10, 5))
    plt.hist(df['age'], bins=20)
    plt.title('Age distribution')
```



0	48.942364	widow/widower	Associate's degree, occupational	False	358	48	22681.539592	45.936
1	46.279085	married	Master's degree	False	736	36	22020.989278	31.0615
2	65.046517	married	Master's degree	True	775	48	22953.000000	34.487
3	31.649470	single	11th grade2	False	828	48	28180.187136	41.666
4	22.286850	married	Some college, no degree	True	789	36	22829.773663	26.974

```
import matplotlib.pyplot as plt

# Plot histogram of 'Age' column from 'marketing' dataframe
plt.hist(marketing['Age'])
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of Age')
plt.show()
```

GitHub Copilot ❌

Einblick ✅

Picked the second out of 5 suggestions. Executing code leads to errors:

- Wrong dataframe name (df instead of marketing)
- Wrong column name (age instead of Age)

The hard stuff nobody talks about

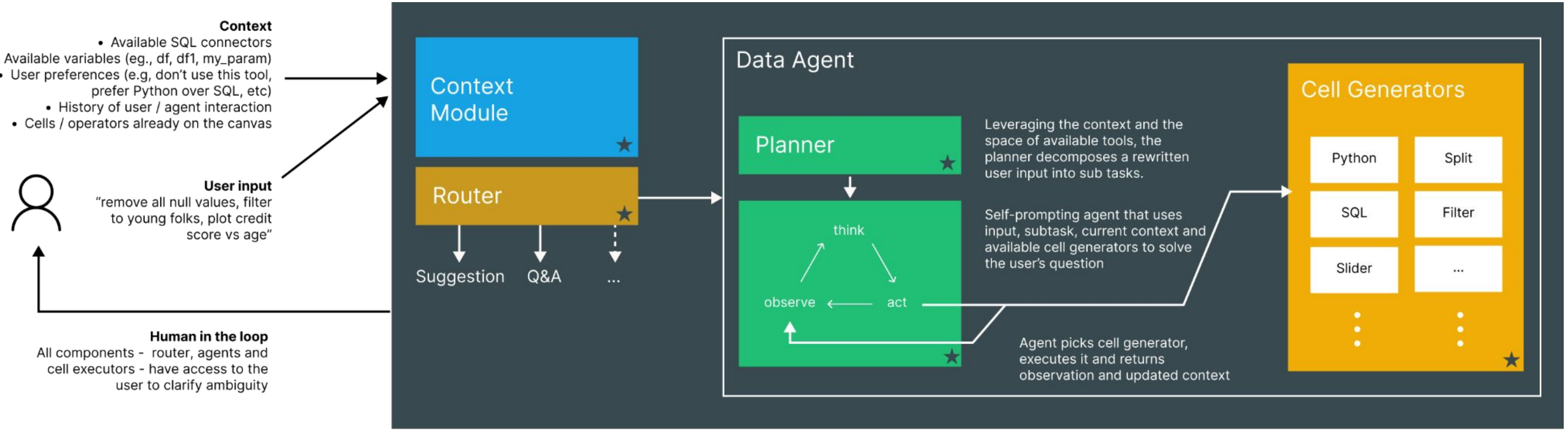
Problem	Solution
LLMs are not a product	Full infrastructure is required, connectors, no-code plugins, Python / SQL execution
Latency latency latency!	Various UI techniques Adaptive routing based on context
Hallucinations	Give LLMs a way out Human in the loop
Testing / Logging ● Small changes have big impact ● Standard engineering techniques fail	Set up extensive infrastructure Automated regression testing using LLMs for comparison
Robustness ● LLMs will return garbage at times	Schema validation, code parsing Self-correction
Multi-modality ● LLMs get confused when mixing languages	Built-in abstraction layer to translate everything in context (e.g., SQL, no-code) to Python
Context ● Limited window size ● Tokens are expensive and slow ● Recency bias	Context module with static and dynamic part Multi hierarchy vector stores Many UI based heuristics

Natural Language Engine Architecture

Context
- Available SQL connectors
- Available variables (eg., df, df1, my_param)
- User preferences (e.g, don't use this tool, prefer Python over SQL, etc)
- History of user / agent interaction
- Cells / operators already on the canvas

User input
"remove all null values, filter to young folks, plot credit score vs age"

Human in the loop
All components - router, agents and cell executors - have access to the user to clarify ambiguity



Leveraging the context and the space of available tools, the planner decomposes a rewritten user input into sub tasks.

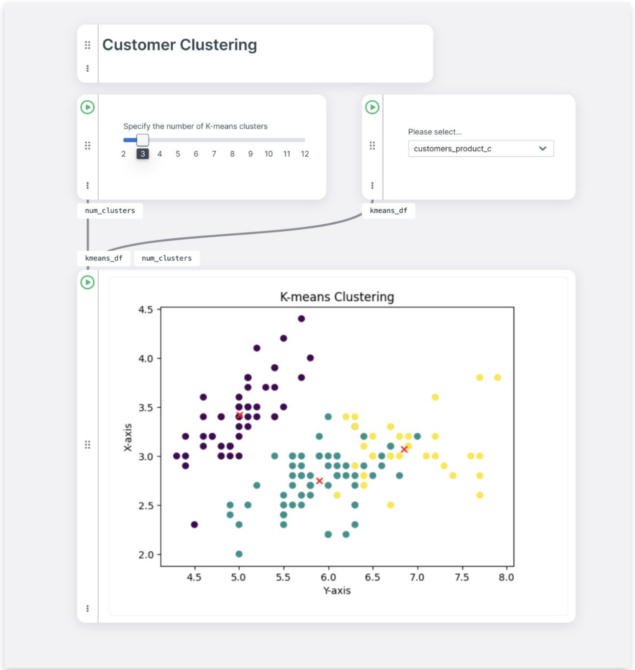
Self-prompting agent that uses input, subtask, current context and available cell generators to solve the user's question

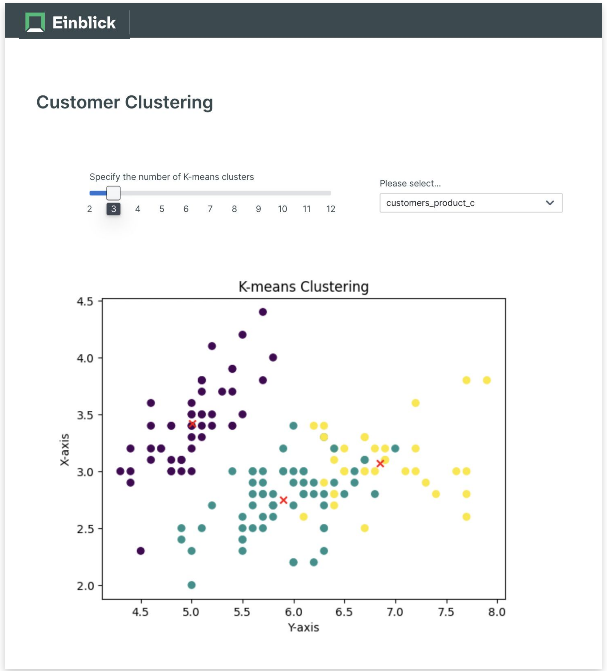
Agent picks cell generator, executes it and returns observation and updated context

★ **LLM Powered**
- All major components within the system are LLM powered
- We currently use OpenAI's gpt-3.5-turbo, however architecture is model agnostic

Data Apps & Plugins

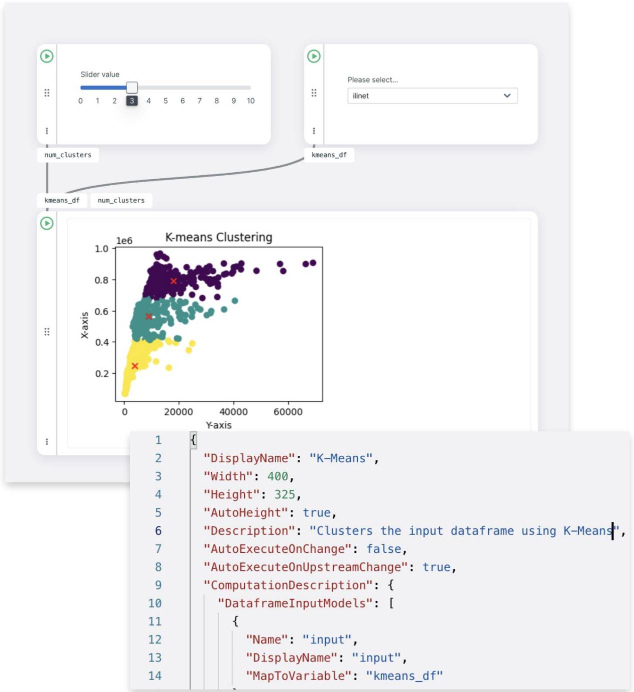
Data Apps



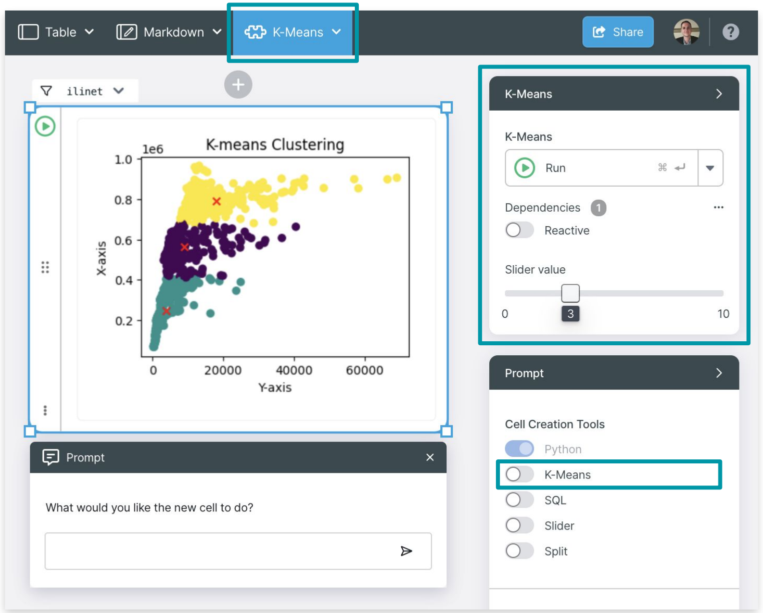


Anything on a canvas can be turned into an **interactive, shareable data app**, with a few clicks. **No code required.**

Plugins



```
{
  "DisplayName": "K-Means",
  "Width": 400,
  "Height": 325,
  "AutoHeight": true,
  "Description": "Clusters the input dataframe using K-Means",
  "AutoExecuteOnChange": false,
  "AutoExecuteOnUpstreamChange": true,
  "ComputationDescription": {
    "DataframeInputModels": [
      {
        "Name": "input",
        "DisplayName": "input",
        "MapToVariable": "kmeans_df"
```



Build or generate plugins **without code**, on the canvas. Plugins are **available as no-code cells** through the menu, and can be made **accessible to the Natural Language Engine**.

www.einblick.ai