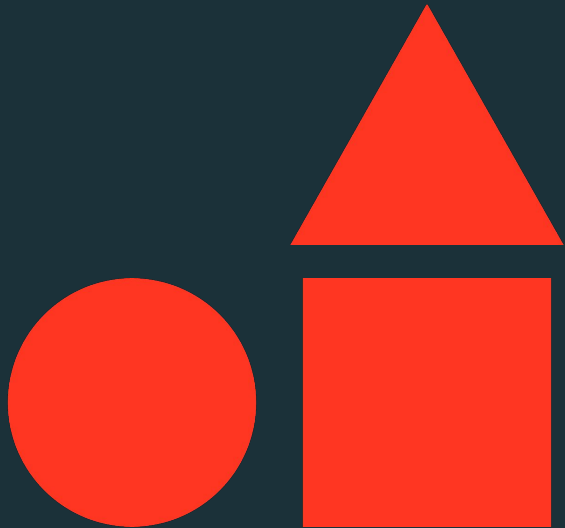


Software Engineering Adapting to the AI Revolution: Thriving in the Age of GenAI

Jules S. Damji

9/25/2025, SVGENAI, Fremont



\$whoami

```
* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/jules/git-repos/pyspark-data-sources





Tips for getting started:

1. Run /init to create a CLAUDE.md file with instructions for Claude
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

* Tip: Start with small features or bug fixes, tell Claude to propose a plan, and verify its suggested edits

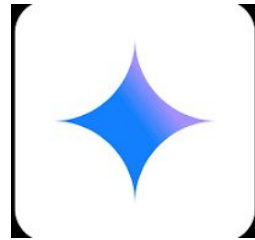
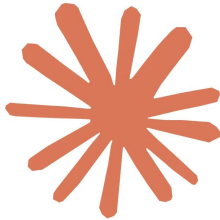
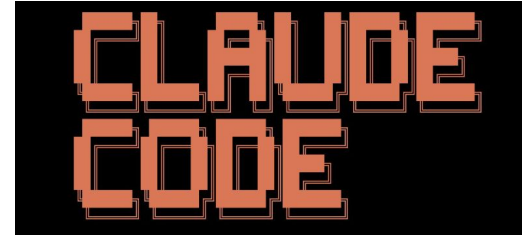
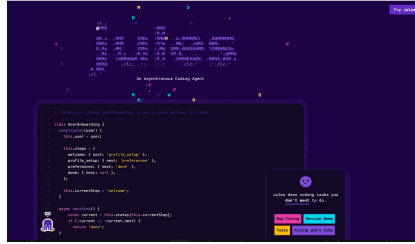
$whoami
└─ Credit balance too low · Add funds: https://console.anthropic.com/settings/billing

> |
```

-  Technical Staff with Developer Advocacy
-  MLflow contributor
-  Co-author *Learning Spark 2nd*
-  Worked at Databricks, Anyscale, Netscape, Opsware/Loudcloud, @Home, Verisign, Sun Microsystems



AI-Assisted Developments



OpenAI Codex



TL; dr

- AI is changing the way we will code
- Collaboration with AI is the new way forward...
- We are in this early ... things are changing rapidly
- Adapt and thrive in the new age ...

**“We are entering the era of
Software 3.0, where natural
language becomes the
programming interface.”**

—Andrej Karpathy

Welcome to the new era!

How software has evolved over the time ...

Software 1.0

computer code

programs

computer



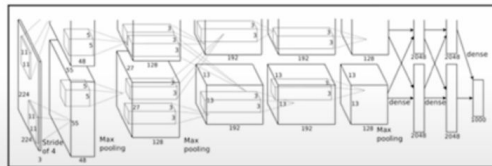
became programmable in ~1940s

Software 2.0

weights

programs

neural net



fixed function neural net

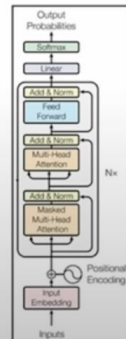
e.g. AlexNet: for image recognition (~2012)

Software 3.0

prompts

programs

LLM



~2019

LLM = programmable neural net!



Welcome to the new era!

How software has evolved over the time ...

Example: Sentiment Classification

Software 1.0

```
python
def simple_sentiment(review: str) -> str:
    """Return 'positive' or 'negative' based on a tiny keyword lexicon."""
    positive = {
        "good", "great", "excellent", "amazing", "wonderful", "fantastic",
        "awesome", "loved", "love", "like", "enjoyed", "superb", "delightful"
    }
    negative = {
        "bad", "terrible", "awful", "poor", "boring", "hate", "hated",
        "dislike", "worst", "dull", "disappointing", "mediocre"
    }

    score = 0
    for word in review.lower().split():
        w = word.strip(",.;!:")
        # crude token clean-up
        if w in positive:
            score += 1
        elif w in negative:
            score -= 1

    return "positive" if score >= 0 else "negative"
```

Software 2.0

10,000 positive examples
10,000 negative examples
encoding (e.g. bag of words)

↓
train binary classifier

↓
parameters

Software 3.0

You are a sentiment classifier. For every review that appears between the tags

<REVIEW> ... </REVIEW>, respond with **exactly one word**, either POSITIVE or NEGATIVE (all-caps, no punctuation, no extra text).

Example 1

<REVIEW>I absolutely loved this film—the characters were engaging and the ending was perfect.</REVIEW>

POSITIVE

Example 2

<REVIEW>The plot was incoherent and the acting felt forced; I regret watching it.</REVIEW>

NEGATIVE

Example 3

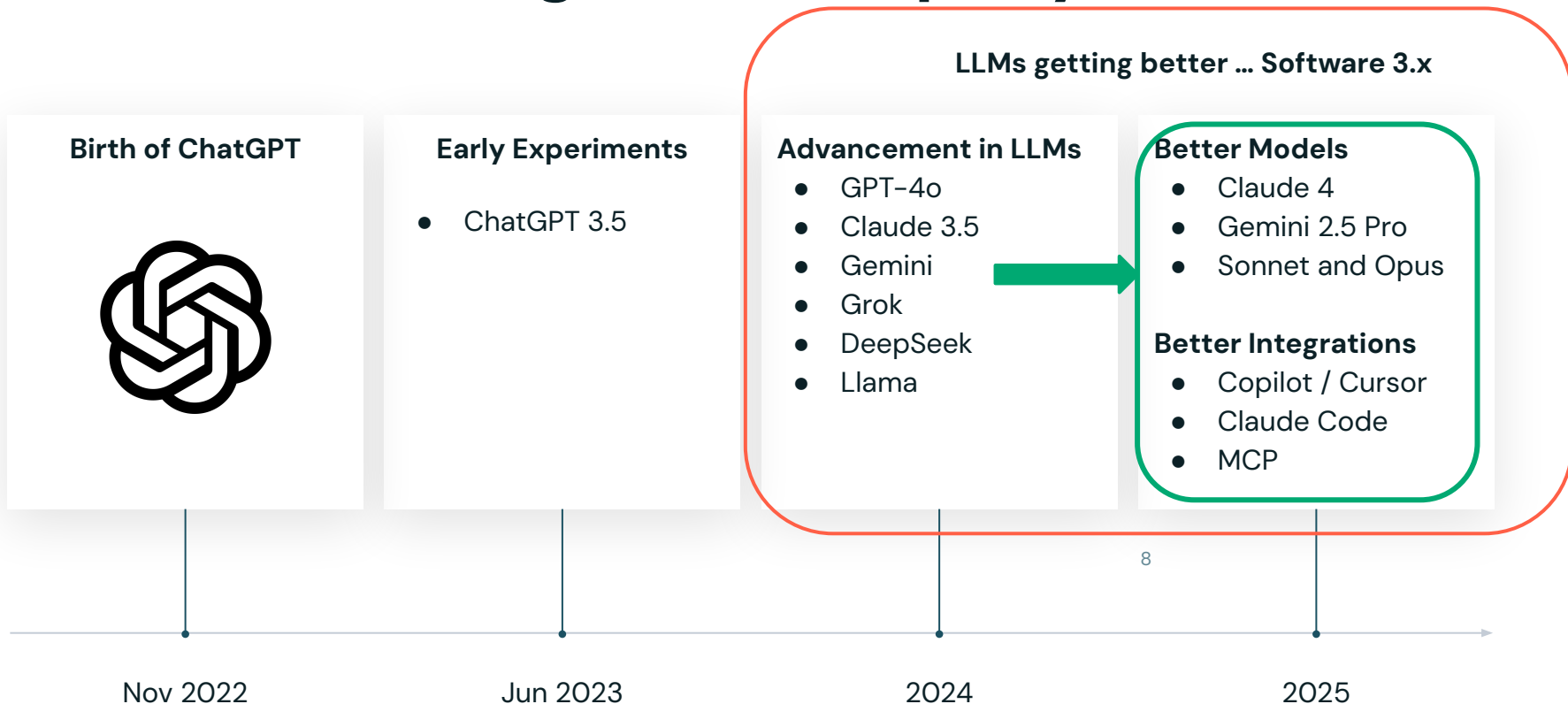
<REVIEW>An energetic soundtrack and solid visuals almost save it, but the story drags and the jokes fall flat.</REVIEW>

NEGATIVE

Now classify the next review.



LLMs Are Getting Better Rapidly



Welcome 🤝 to the new Era: Software 3.x!

- Software engineering is undergoing a seismic shift
- Gen AI & Agentic AI are redefining workflows and how we incrementally build software
- *Change brings both uncertainty and opportunity*



We are seeing **bullish** predictions of code generation with Software 3.x..

News By [Hassam Nasir](#) published April 30, 2025

 Comments (16)



Inc.

NEWSLETTERS

SUBSCRIBE

 $\rho = \rho_0$

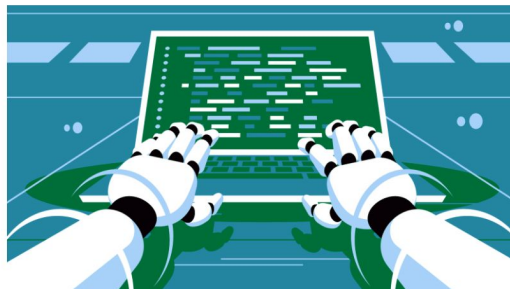
YOUR NEXT MOVE

AI-Generated In a Year It's time to stop taking tech zealots seriously.

EXPERT OPINION BY JOE PROCOPIO, FOUNDER, JOEPROCOPIO.COM #JPROCO

MAR 20 2025

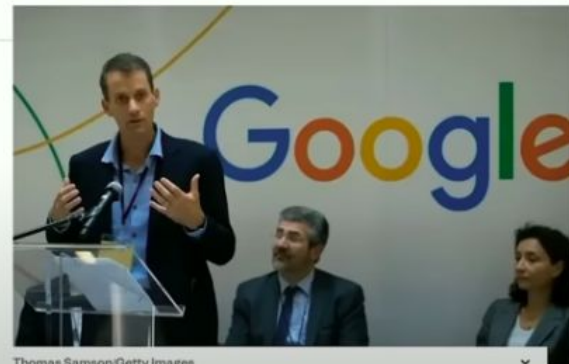
SHARE



AI

Google chief scientist predicts AI could perform at the level of a junior coder in a year

By Sarah Perkel



Thomas Samson/Getty Images

What is the reality?

Pragmatic engineer surveyed adoption patterns across the Industry...

1. AI Dev tools startup
2. Big Tech
3. AI Startups
4. Individual Software Engineers



Adoption Patterns Across Industry

A TL;Dr of Pragmatic Engineer's findings ...

- **AI Startups:** High adoption and fast experimentation with LLMs
- **Big Tech:** Cautious, infrastructure-focused integration
- **Independent Engineers:** Growing enthusiasm and creativity



Anthropic



“We are only scratching the surface of what’s possible. The model is moving really fast—it’s exponential and it’s getting better at coding very, very, quickly.”

–Boris Cherny, Anthropic, Creator of Claude Code

“When we gave Claude Code access to engineers, they all started using it everyday, which was pretty surprising...”

“90% of the Claude Code is written with Claude Code”

–Anonymous engineers, Anthropic



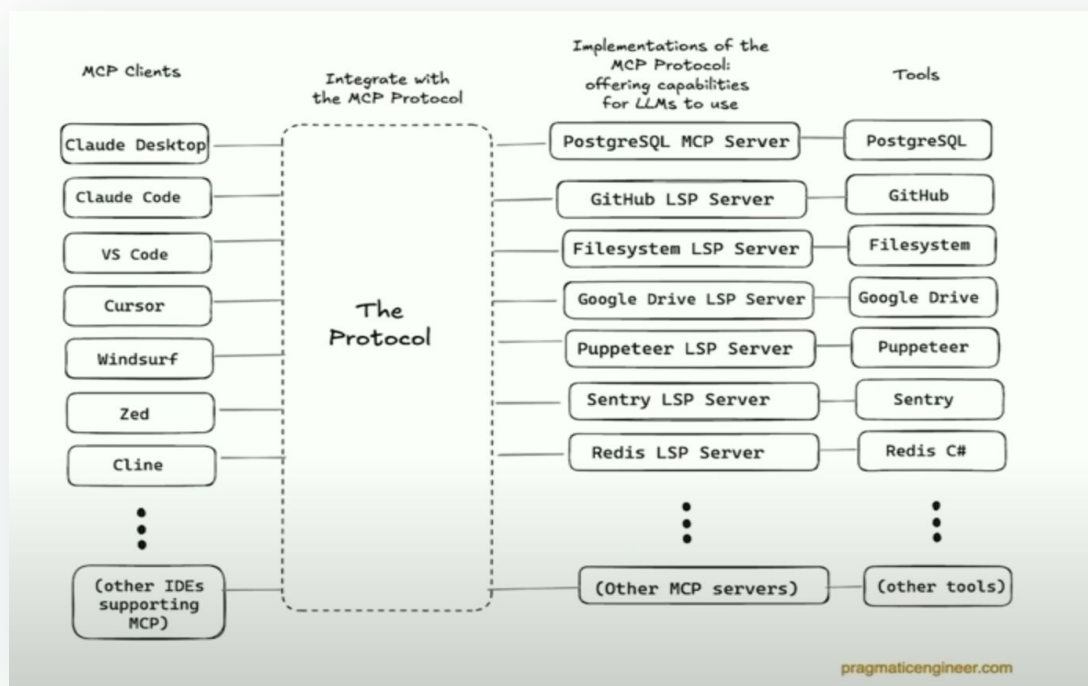
Claude Usage 🚀

- Claude Code usage climbed 🪜 by 40% on day of release
- User base grew 160% 📈 increase since launched
- Now part of onboarding for new hires

Model Context Protocol

Facilitated and extended Agent's capabilities: tools, resources, prompts

An Agents Gateway to the world:



OpenAI

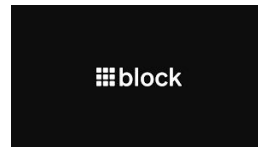
Google

Microsoft



replit

Sourcegraph



Windsurf

Seasoned Developers



Armin Ronacher

Creator of Flask, Jinja2, and Founder at Sentry



AI Changes Everything ...

A moment of reflection & epiphany ...



[Armin Ronacher](#)'s Thoughts and Writings

[blog](#) [archive](#) [tags](#) [projects](#) [talks](#) [about](#)

AI Changes Everything

written on Wednesday, June 4, 2025

At the moment I'm [working on a new project](#). Even over the last two months, the way I do this has changed profoundly. Where I used to spend most of my time in Cursor, I now mostly use [Claude Code](#), almost entirely hands-off.

Do I program any faster? Not really. But it feels like I've gained 30% more time in my day because the machine is doing the work. I alternate between giving it instructions, reading a book, and reviewing the changes. If you would have told me even just six months ago that I'd prefer being an engineering lead to a virtual programmer intern over hitting the keys myself, I would not have believed it. I can go can make a coffee,

“I currently don’t sleep a lot.”

“ Claude Code is a catnip for programmers...and it feels like it.... ”

“I have never felt so energized, confused, willing to try new things.... It’s incredibly very addicting”

AI Changes Everything ...

A moment of confession & acceptance ...

- “Claude Code got shockingly good” 😊
- “By using LLMs extensively I got over the ‘hurdle’ of not accepting them
- “The fault of the model are avoided because now the tool can run the code and see what happens”

Simon Willison

Creator of Django

AI Changes Everything ...

Slogging at blogging LLMs...



Andrej Karpathy ✓

@karpathy



Congrats to Simon Willison (@simonw) on 23 years (!!) of blogging. Really excellent LLM blog, I sub & read everything:

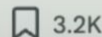
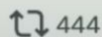
simonwillison.net

(e.g. I sub via RSS/Atom on NetNewsWire)

+If you consistently enjoy the content like I do, sponsor on GitHub:

github.com/sponsors/simonw

7:48 PM · Jun 13, 2025 · **582.1K** Views



**“Coding agents are a thing that actually work now:
run an LLM in a loop, let it execute compilers and
tests and linters and other tools, give it a goal and
watch it do the work for you.**

**The model improvements in the last six months have
tipped these over from fun toy demos to useful on
daily basis”**

Consider as Cooperation with *partial autonomy*

Trust but verify ...

Consider the full workflow of partial autonomy UIUX

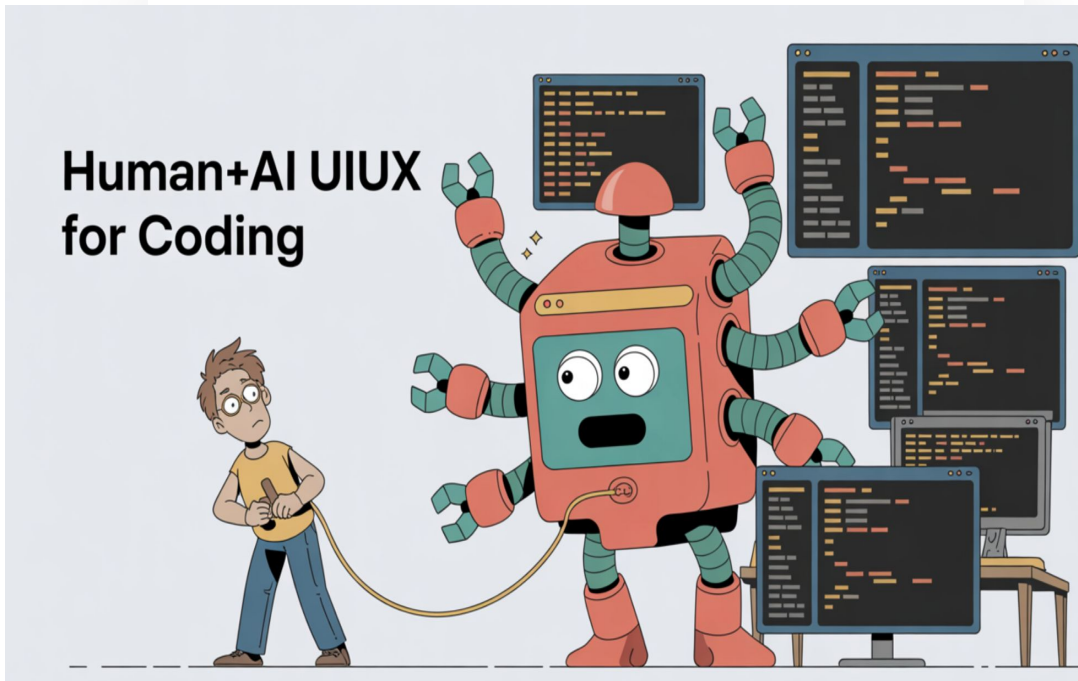


Partial Autonomy

For successful verification

Start with concrete tasks

- Describe single, concrete steps
- Ask approaches
 - pick an approach, draft code
- review/learn: pull up API docs, ask explanations
- Test and verify
- Ask what next

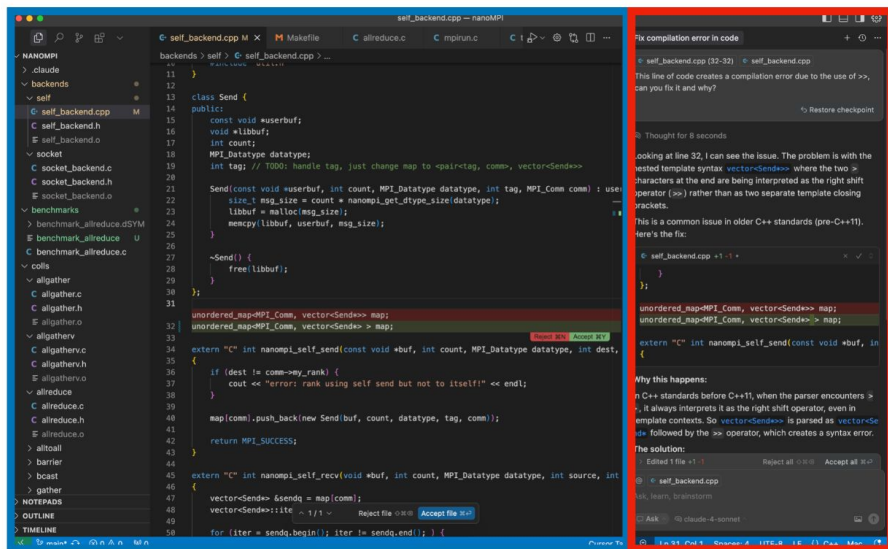


An example of Generation & Verification

Example: Anatomy of Cursor

Traditional interface

LLM integration



1. Package state into a context window before calling LLM.
2. Orchestrate and call multiple models (e.g. embedding models, chat models, diff apply models, ...)
3. Application-specific GUI
4. Autonomy slider: Tab → Cmd+K → Cmd+L → Cmd+I (agent mode)

autonomy slider

An example of Generation & Verification Cursor + Claude Code: *partial autonomy*

Traditional interface

LLM Integration

The screenshot displays the Cursor IDE interface with Claude Code integration. The left sidebar shows a file explorer with the following structure:

- ETL-PIPELINES
 - src.py
 - generators
 - ldp
 - music_analytics
 - images
 - transformations
 - ldp_musical_pipeline.py
 - README.md
 - __init__.py
 - sdp
 - gignore
 - CLAUDE.md
 - README.md

The main editor displays the content of `musical_pipeline.py`, which includes a README-style overview and a diagram of the Medallion architecture. The right pane shows the Claude Code chat interface with a prompt about adding benefits to the README and a response listing three benefits. A red arrow points to the "Opened changes in Cursor" button at the bottom of the chat.

musical_pipeline.py

```
1 # Music Analytics - Databricks Lakeflow Declarative Pipeline
2
3 ## Overview
4
5 This directory contains a Lakeflow Declarative Pipeline (LDP) implementation for processing and analyzing the Million Song Dataset. The pipeline leverages Apache Spark and Lakeflow's declarative transformations with streaming data ingestion and comprehensive data quality validation.
6
7 ## Key Benefits of Lakeflow Declarative Pipelines
8
9 1. Simplified Development & Maintenance: LDP uses declarative decorators that automatically handle complex orchestration, dependency management, and error recovery, reducing code complexity by up to 80% compared to traditional ETL frameworks.
10
11 2. Built-in Data Quality & Governance: Native @elt.expect decorators provide comprehensive data validation with automatic quarantine of bad records, detailed quality metrics, and lineage tracking without additional infrastructure setup.
12
13 3. Auto-scaling & Cost Optimization: LDP automatically optimizes cluster sizing, manages incremental processing, and provides intelligent caching strategies, reducing compute costs by 40-60% while maintaining sub-second query performance on large datasets.
14
15 ## Pipeline Architecture - Medallion Pattern
16
17 ARCHITECTURE
18
19 Lakeflow Declarative
20
21 Pipelines
22
23 BRONZE LAYER (Raw Data)
24
25 songs_raw_bronze
26
27 Raw CSV ingestion via Auto
```

Claude Code

Silver Layer Enhancement:

- Added artist_location to songs_metadata_silver table
- Added validation: @elt.expect("valid_artist_location", "artist_location IS NULL OR trim(artist_location) = """)

Gold Layer Fix:

- Changed data source: Now reads from dlt.read("songs_metadata_silver") instead of bypassing silver layer
- Updated aggregations: Replaced avg_tempo_bpm (not available in metadata silver) with unique_artists from location (more relevant for location analysis)
- Added proper validation:
 - @elt.expect("valid_location_summary", "location IS NOT NULL AND trim(location) = """)
 - @elt.expect("positive_song_count", "songs_from_location > 0")
 - @elt.expect("positive_artist_count", "unique_artists_from_location > 0")
 - @elt.expect("reasonable_duration_location", "avg_duration_seconds > 0 AND avg_duration_seconds < 3600")

Benefits:

- Proper Data Lineage: Bronze - Silver - Gold ✓
- Data Quality: Comprehensive validation at both silver and gold layers ✓
- Architecture Consistency: All gold tables now read from silver tables ✓
- Better Analytics: Added artist count by location for richer insights ✓

The artist_location_summary_gold table now properly follows LDP best practices and maintains the integrity of the medallion architecture!

You can add three benefits about Lakeflow declarative pipelines and its approach after the overview in the README file! Also add those benefits in the opening comments in ldp_musical_pipeline.py. There is also a reference to ldp_songs_dataset.py in that file remove it.

- I'll add the three benefits about Lakeflow Declarative Pipelines to both files and clean up the incorrect filename reference.
- First, let me add the three LDP benefits after the overview in the README:
- Update(src/py/ldp/music_analytics/README.md)

Todos:

- Add three LDP benefits after overview in README.md
- Add three LDP benefits to opening comments in ldp_musical_pipeline.py
- Remove incorrect ldp_songs_dataset.py reference from pipeline file

Opened changes in Cursor

Save file to continue.

Do you want to make this edit to README.md?

- > Yes
- > Yes, allow all edits during this session (shift+tab)
- > No, tell Claude what to do differently (esc)

“2025–2035 is the decade of agents”

—Andrej Karpathy

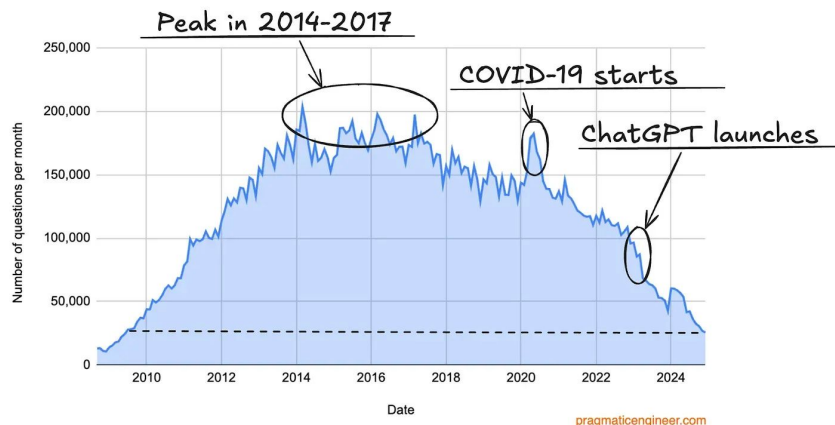


Apache Spark, Gen AI & Agentic AI

Paradigm Shift in Software Development

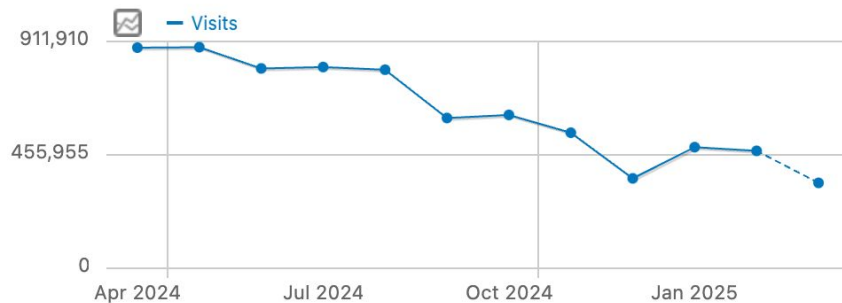
Developers are moving away from forum and docs

Monthly questions asked on StackOverflow

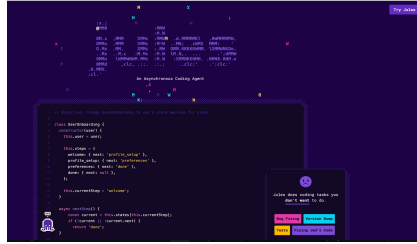


Apache Spark Doc Website Visits

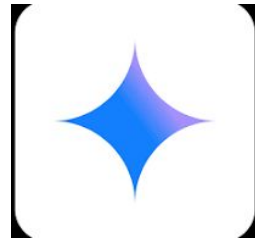
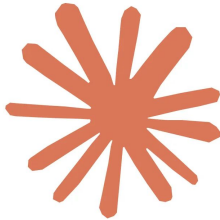
Visits Over Time



AI-Assisted Developments



OpenAI Codex



CURSOR



LLM friendly developer docs

GitIngest

NEW /llm.txt GitHub 10.9k



Prompt-friendly codebase



Turn any Git repository into a simple text digest of its codebase.
This is useful for feeding a codebase into any LLM.



Exclude ▾ *md, src/

Include files under: 50kB

☐ Private Repository NEW

Try these example repositories:

Gitingest

FastAPI

Flask

Excalidraw

ApiAnalytics

You can also replace 'hub' with 'ingest' in any GitHub URL.

Summary

Repository: apache-spark/spark
Files analyzed: 10000
Estimated tokens: 5.8M

Directory Structure

Directory structure:
└─ apache-spark-spark/
 ├─ LICENSE
 ├─ make-distribution.sh
 ├─ NOTICE
 ├─ pom.xml
 ├─ scalastyle-config.xml
 ├─ tox.ini
 └─ .rat-excludes

Files Content

(Files content cropped to 300k characters, download full ingest to see more)
FILE: LICENSE
=====

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

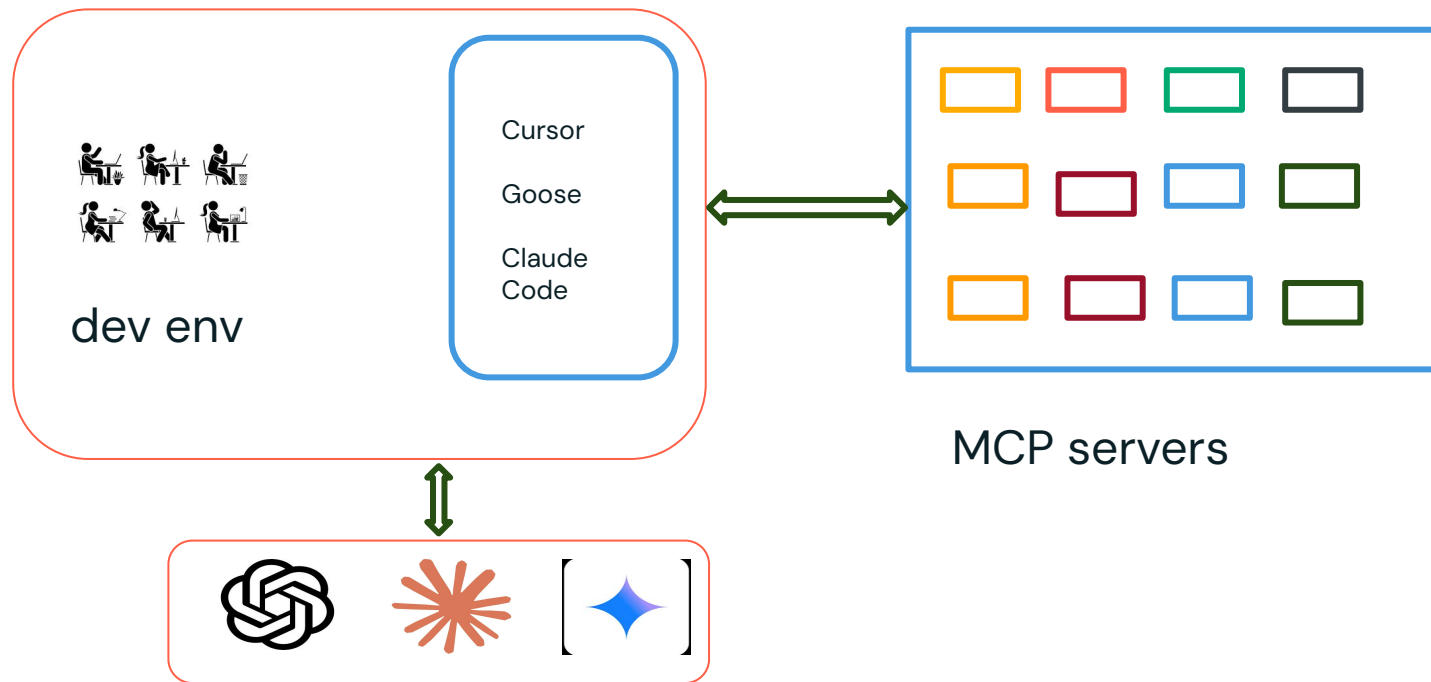
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.



Experimentation: Cursor, MCP, Goose, Claude Code



Experimentation and Use cases for AI tools

- How is Python `DataSource` class defined and used?
- Explain what public functions I need to implement in the derived class
- Summarize `JIRA-XXX` with PR `#12345`?
- Review suggested changes in the PR `#3456`
- Suggest how to resolve changes in the PR `#3456`
- Show me how can I do `X` to extend Spark SQL function `fx()`
- Can you explain the data flow between the `Spark Connect client` and `Spark Connect Server`?
- Explain the code in `DataSource.py`



A You are an expert in Apache Spark and PySpark internals. Generate a Python Data Source to fetch daily news using the NewsAPI using the PySpark Data Source API.

I'd be happy to create a PySpark Data Source that fetches daily news from the NewsAPI. This involves implementing the necessary reader/writer interfaces for custom data sources in Spark.

Let me create this for you using the PySpark Data Source API:

NewsAPI PySpark Data Source
Code



I've created a comprehensive PySpark Data Source implementation for fetching news from the NewsAPI. Let's go through the key components and how to use it:

Key Components

1. **NewsAPIClient**: A client class that handles HTTP requests to the NewsAPI endpoints with proper error handling.
2. **NewsAPISource**: The main implementation that transforms API responses into DataFrame-compatible data.
3. **DefaultSource** and **NewsAPIRelation**: Classes required for the Spark Data Source API V2 integration.

class NewsAPISource:

"""

Implementation of PySpark's Data Source API for NewsAPI.

"""

@staticmethod

def _extract_articles(api_response: Dict[str, Any]) -> List[Dict[str, Any]]: """Extract and transform articles from the API response."""

articles = []

for article in api_response.get("articles", []):

processed_article = {

"source_id": article.get("source", {}).get("id"),

"source_name": article.get("source", {}).get("name"),

"author": article.get("author"),

"title": article.get("title"),

"description": article.get("description"),

"url": article.get("url"),

"url_to_image": article.get("urlToImage"),

"published_at": article.get("publishedAt"),

"content": article.get("content")

}

articles.append(processed_article)

return articles

@classmethod

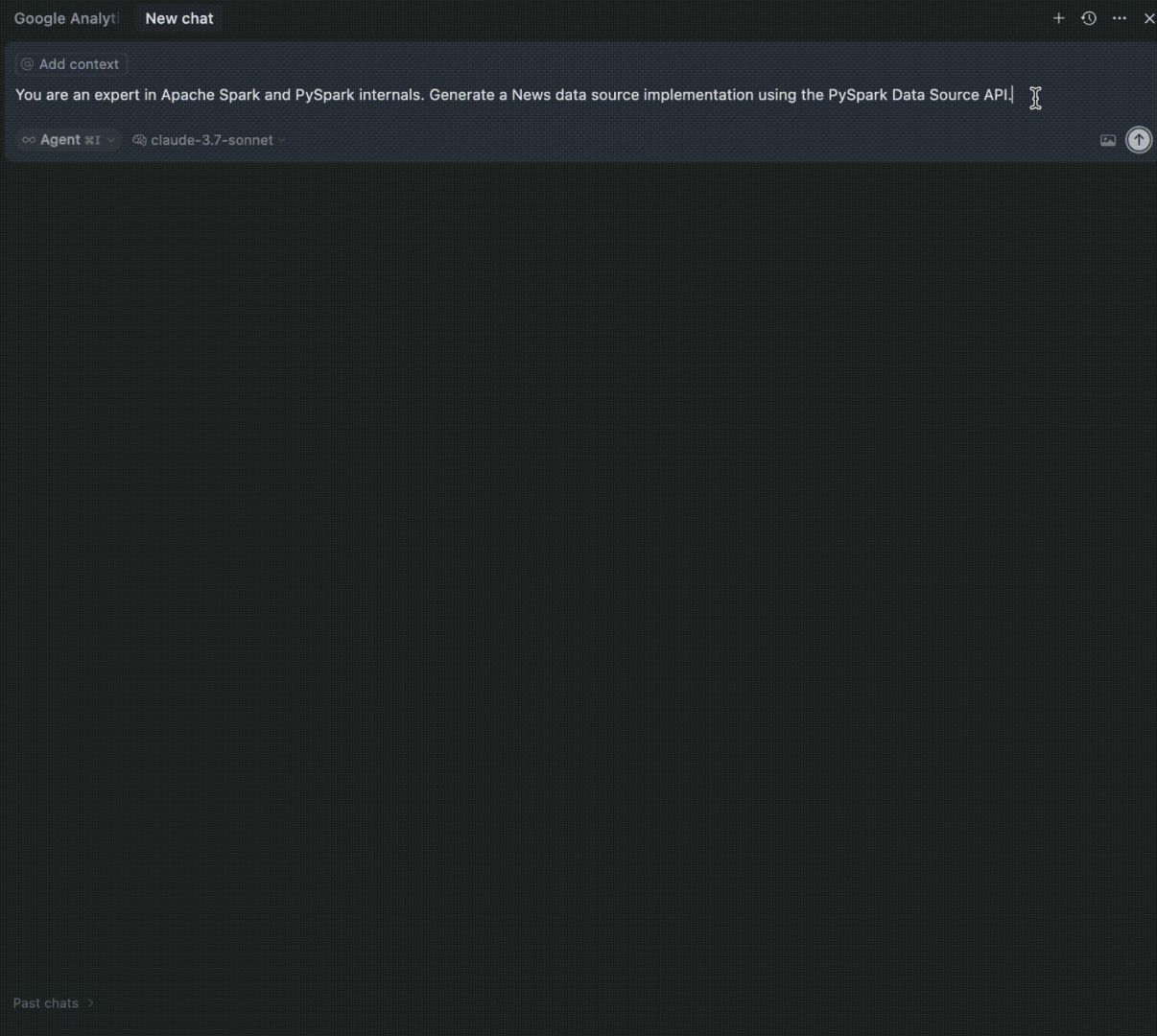
def read_news(cls, spark: SparkSession, options: Dict[str, str]) -> Data

"""

Fetch news articles based on provided options and create a DataFrame

Asking Cursor to
generate a new Python
Data Source given the
example repo as context:

[pyspark-data-sources](#)





Takeaways

- AI is changing the way we will code
- Collaboration with AI is the new way forward...
- We are in this early ... things are changing rapidly
- Adapt and thrive in the new age ...

Upcoming events of interest for SVGENAI ...



The banner features a dark blue background with a grid of white dots and lines. On the right side, there are several overlapping squares in shades of orange and blue. At the top left, the Apache Spark logo (a red star) and the NVIDIA logo (a white eye) are displayed. The main title is in large white font, and the date is in a smaller white font. Below the title, there are three circular headshots of the speakers, each with their name and title in white text below them.

APACHE Spark **NVIDIA**

Spark Connect: NVIDIA Accelerator for Spark SQL and MLlib

October 29 @ 9:30 AM PST



Jules S. Damji
Staff Developer Advocate
Databricks



Erik Ordentlich
Senior Manager
NVIDIA



German "Gera" Shegalov
Principal System Engineer
NVIDIA

Upcoming events of interest for SVGENAI ...


Upcoming Events

[October 7] Webinar | From Functions to AI Agents:
Reimagining the Lakehouse for an Agentic Future

[November 13] Open Lakehouse + AI Mini Summit | Mountain
View

- Followed up an Apache Spark Happy Hour!

[November 24] Open Lakehouse + AI Meetup | Paris

Learn more & RSVP at luma.com/open-lakehouse 



It's Early 🕒 , Embrace 😊 the shift,
Experiment 🔬 , Empower 👩💻
with 🤖

Thank you!

