Evaluation of LLM Analysis on Pandas Github Data

1. Key Finding

While the engine successfully identified sentiment patterns and specific soft-skill strengths, it failed to generate "Growth Opportunities." Our network analysis confirms this is due to the structural "sparsity" of open-source collaboration (1.1% density) compared to the dense nature that would be expected enterprise teams.

2. Introduction

2.1 Objective

The ClarityLoop engine is designed to identify coaching moments within corporate teams. This run stress-tested that logic to determine:

1. **Transferability:** Can an enterprise model correctly interpret the tone and intent of open-source code reviews?
2. **Limitations:** What structural differences prevent the engine from identifying developmental feedback?

2.2 Dataset Selection

The `pandas-dev/pandas` repository was targeted due to its scale, professional standards, and active maintainer community. To ensure high-quality data, a "Long-Term Contributor" filter was implemented, isolating users who have been consistently active for at least **5 consecutive years**. This filtered out "drive-by" contributors and focused on the core community that most closely resembles a professional team.

3. Methodology

3.1 Data Gathering Pipeline

Developed a Python pipeline (`clarityloop/collabsense/src`) to mine GitHub API data. The pipeline consists of:

1. **Async Scraper:** A multi-token, asynchronous scraper capable of handling GitHub's rate limits to fetch thousands of Pull Requests and Comments.
2. **Processor:** A logic engine that filters data based on user tenure (Year-over-Year consistency) and interaction quality (minimum comment thresholds).
3. **Anonymizer:** A cleaning module that replaces real identities with realistic synthetic profiles (Faker), also creating other fake data such as email addresses, ethnicity, etc. to simulate a private enterprise workspace.

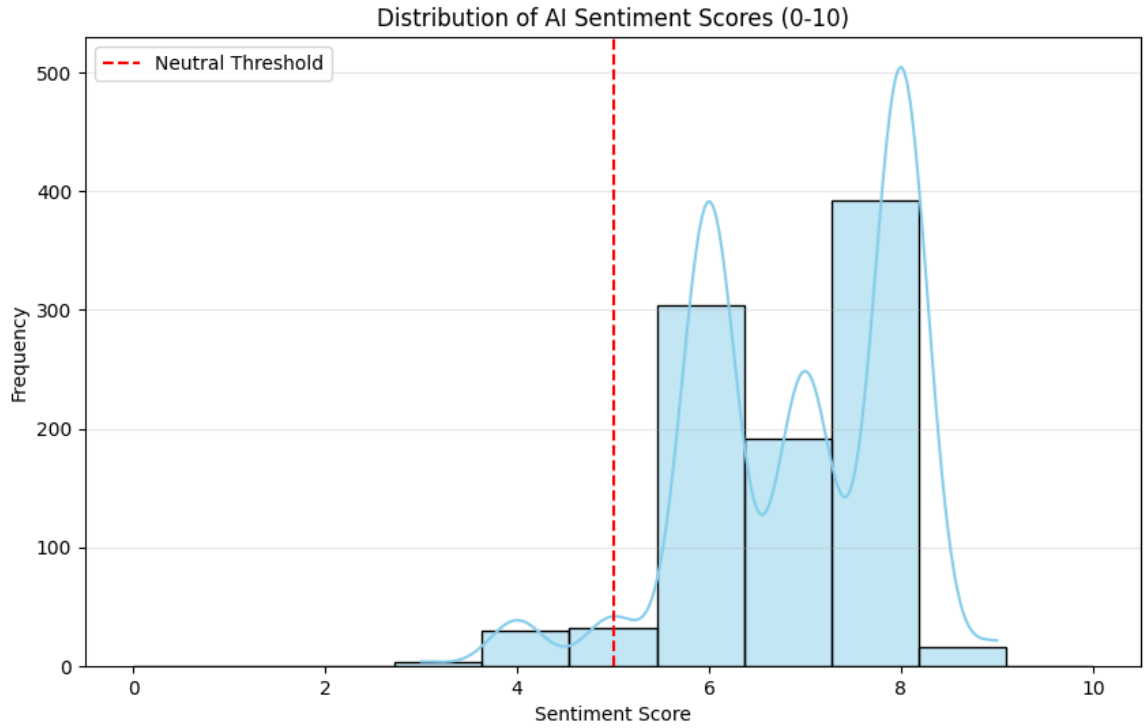
3.2 Analysis Engine

The mined data was ingested into a local instance of the ClarityLoop engine. The raw SQL tables (`comment`, `strength`, `user_info`) were extracted to perform statistical and network analysis.

4. Results & Evaluation

4.1 Sentiment Analysis (High Accuracy)

The engine demonstrated a great understanding of technical collaboration. Rather than clustering all interactions as "Neutral", the sentiment scores formed a distribution with peaks at **6 (Constructive/Neutral)** and **8 (Positive/Praising)**.



Interpretation: This distribution suggests the AI is not "hallucinating" positivity. It distinguishes between transactional acknowledgments (merges, minor fixes) and genuine appreciation. The lack of low scores (0-4) likely reflects the high professionalism of the `pandas` core team rather than a model failure.

Verification: To verify this, several comments should be reviewed manually.

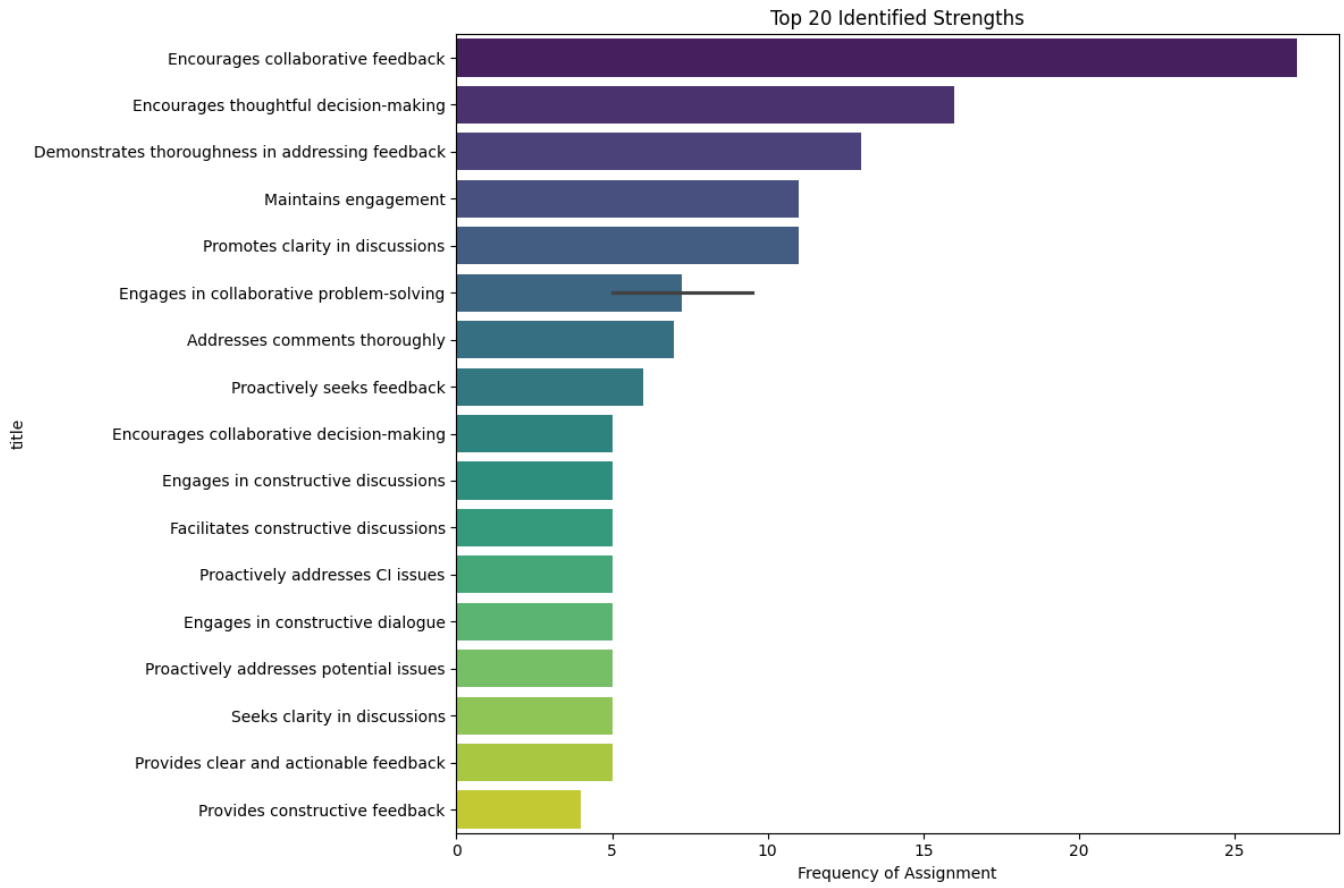
Score	User	Comment Excerpt	Link
4	Simon Hawkins	*"Closing as needs info. Can you post an example that is copy-pastable without having to load an unknown zip file? See https://matthewrocklin.com/minimal-bug-reports.html and we can re-open..."*	View
4	Kristin Macias	*"> I read the above as using PyArrow types by default in 3.0\r\n\r\nI don\t think I've seen that seriously considered/discussed anywhere. There's a path to making it feasible to use pyarrow types by default..."*	View
4	Marc Garcia	*"Thanks for the update here @tqa236. The fix you implemented seems a hack, not the proper fix. You should identify what\s causing the exception..."*	View

Score	User	Comment Excerpt	Link
5	Richard Shadrach	*"Ah, this is a duplicate of https://github.com/pandas-dev/pandas/issues/48476 which is on our roadmap, e.g. https://github.com/pandas-dev/pandas/pull/58988. Closing...."*	View
5	Richard Shadrach	*"I would not expect the keys to be suffixed, especially on an inner join...."*	View
5	Marc Garcia	*"@jbrockmendel while in general I agree with your point, we already have the numba engine in pandas. Do you think we should remove it?\n\nFor what I understand, seems like Bodo should be better for most users, as it works with Arrow types (besides the other advantages discussed). So, while I\m a big fan of not adding more things into pandas..."*	View
6	Joris Van den Bossche	*"> But I do think a more informative error message here is zoneinfo\s responsibility (and would help all Python users..."*	View
6	Richard Shadrach	*"From tzdata docs\n\n> It is generally recommended that any time zone libraries should attempt to use the system data before using the tzdata package, but some systems (notably Windows) do not deploy zoneinfo binaries of this type, and so tzdata is necessary.\n\nIt seems to me we should follow this recommendation if there is a way to check that everything will work as expected using the system data. But I too am curious how one is installing pandas without getting tzdata...."*	View
6	Kristin Macias	*"doing this inside the khash code is definitely more difficult, but problably the Right Way To Do It. Does this entail a perf hit?\r\n\r\nBTW #62888 is probably going to have to entail digging into that same bit of khash code...."*	View
7	Matthew Roeschke	*"Thanks for the PR, but it appears another contributor fixed this area in the meantime so closing. Happy to have a similar contribution to `zip` calls that haven\t been addressed yet'..."*	View
7	Richard Shadrach	*"Thanks for the report. It would be helpful to know exactly what tests you\re seeing so we can be more confident we're fixing them all.'..."*	View
7	Richard Shadrach	*"Thanks for the report. This was added in https://github.com/pandas-dev/pandas/pull/39486. The issue is that these dimensions can be unreliable, so to improve the robustness of pandas we instruct openpyxl to ignore them. I agree that having this impact the user\s handle is a downside and desired to fix..."*	View
8	Richard Shadrach	*"Thanks for the report. If one has a unique index, then `df.loc[df.index[n], \"foo\"]` is also an option. But if index and columns are not unique, then I think you need to find the index you want to change by integer and use `.iloc`. \n\nI\m positive on adding this to the docs..."*	View

Score	User	Comment Excerpt	Link
8	Joris Van den Bossche	*"> One unintended side effect is that `pd.Series([np.timedelta64(1, \"M\")])` raises instead of casting to "30 days 10:29:06". Need to give that some thought.\r\n\r\nMy feeling is that this is a positive change..."*	View
8	William Ayd	*"I think Matt is on vacation so let\s merge for now. Thanks @jbrockmendel '..."*	View
9	Joris Van den Bossche	*"@mpage thanks a lot for taking a look and fixing it for the example! (will try to test it for pandas tomorrow)\n\n> We can also implement `is_local_in_caller_frame` in pure Python:\n\nThat\s even easier. I can test the performance difference to see it it is worth doing in C. Apart from that..."*	View
9	Richard Shadrach	*"Thanks for all your contributions - and pandas and everything else!..."*	View
9	Matthew Roeschke	*"Thanks for all the great contributions to pandas, @MarcoGorelli! Happy to have you back at any time ? but I know you\re working on great things in the ecosystem ? '..."*	View

4.2 Strengths Identification (Moderate Success)

The engine successfully identified specific soft skills.



Top Identified Strengths:

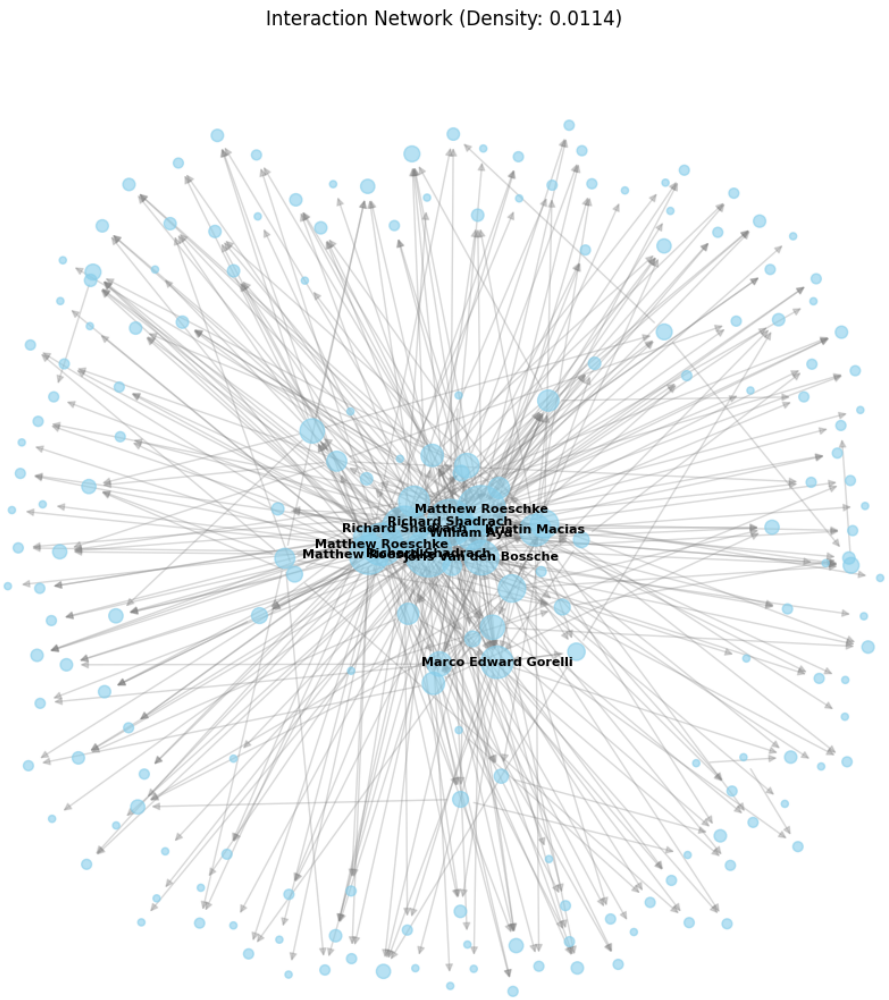
1. **Encourages collaborative feedback** (Frequency: 27)
2. **Encourages thoughtful decision-making** (Frequency: 16)
3. **Demonstrates thoroughness** (Frequency: 13)

Observation: The prevalence of "Encourages collaborative feedback" as the #1 strength validates the engine's ability to identify leadership behaviors even in a flat, open-source structure.

4.3 Growth Opportunities (The "Gap")

The most significant thing was the complete absence of generated Growth Opportunities. The engine returned zero results for this category.

To explain this failure, analysis of the **Network Density** for interactions was done. Enterprise environments typically feature dense clusters where a manager and peers review/leave feedback for the same employee repeatedly. Open Source is characterized by "Sparsity."



Network Metrics:

- **Active Users (Nodes):** 197
- **Feedback Loops (Edges):** 442
- **Network Density: 0.011 (1.1%)**
- **Reciprocity:** 0.10 (10%)

TODO: confirm what the actual parameters are/what values and conditions are needed to generate a growth opportunity

Conclusion: The graph illustrates a few central maintainers interacting with many disconnected contributors. The network density of **1.1%** confirms that open-source collaboration is highly transactional (i.e. out of every 100 possible pairs of people who could be talking to each other, only one pair actually is, as almost everyone is a stranger to one another. in a real team it should be much higher density). A growth opportunity algorithm designed to trigger on "recurring patterns of feedback from the same person" will statistically *never* fire in a network this sparse.

TODO: from mondays meeting, change the LLM prompt so it always gives growth opportunities and re-run, pssibly with smaller dataset just for testing

5. Discussion: Enterprise vs. Open Source

The lack of Growth Opportunities is not a failure of the engine's *comprehension*, but a mismatch in *trigger conditions*.

Feature	Enterprise Team	Open Source Project
Relationship	Manager-Employee (Long term)	Maintainer-Contributor (Transactional)
Feedback Goal	Career Growth & Coaching	Code Quality & Merging
Density	High (Dense clusters)	Low (Sparse / Disconnected)

The AI is tuned to look for the left column (from my current understanding). When fed data from the right column, it correctly identifies that no "Managerial Coaching" patterns exist.

6. Recommendations & Future Work

For non-enterprise deployments like in this test, the "trigger threshold" or Growth Opportunities needs be lowered. Since enterprise deployments *are* the target, this is not a priority, so instead I must find a way/data source other than pandas that is closer to an actual workplace.

But in this specific case the current requirement for **pattern recognition** (frequency/density) seems too strict.

6.2 Alternative Data Sources

Future research should investigate datasets that better mimic the "Teacher-Student" or "Manager-Employee" dynamic:

- **SmartSHARK:** Investigation confirms that this dataset consists exclusively of open-source Apache projects. While extensive, it reflects the same sparse, transactional social structure as the pandas

repository and therefore may not be a suitable replacement to get enterprise-like data.
- **University Coursework:** Repositories where students receive consistent, grading-focused feedback from TAs would provide the density required to test the Growth Opportunity logic.
- **Literature Review:** To see if any similar work has been done since the year or so ago last checked.

7. Conclusion

This study confirms that while the LLM-based analysis can easily generalise **Sentiment** and **Strengths** across domains, higher-order insights like **Growth Opportunities** are deeply dependent on the social structure of the data. The 1.1% network density of open-source projects is insufficient to trigger coaching patterns without modification to the engine's sensitivity.